NAVAL POSTGRADUATE SCHOOL
MONTEREY, CALIFORNIA

THESIS

DEVELOPMENT OF ONBOARD DATA ACQUISITION FOR UNMANNED AIR VEHICLE FLIGHT TESTING

by

Joseph M. Merola

December 1996

Thesis Advisor: Richard M. Howard

Approved for public release; distribution is unlimited

REPORT DOCUMENTATION PAGE

Form Approved OMB No. 0704-0188

1. AGENCY USE ONLY *(Leave blank)*	2. REPORT DATE December 1996.	3. REPORT TYPE AND DATES COVERED Master's Thesis

4. Development of Onboard Data Acquisition for Unmanned Air Vehicle Flight Testing

5. FUNDING NUMBERS

6. AUTHOR(S) Joseph M. Merola

7. PERFORMING ORGANIZATION NAME(S) AND ADDRESS(ES)
Naval Postgraduate School
Monterey CA 93943-5000

8. PERFORMING ORGANIZATION REPORT NUMBER

9. SPONSORING/MONITORING AGENCY NAME(S) AND ADDRESS(ES)
N/A

10. SPONSORING/MONITORING AGENCY REPORT NUMBER
N/A

11. SUPPLEMENTARY NOTES The views expressed in this thesis are those of the author and do not reflect the official policy or position of the Department of Defense or the U.S. Government.

12a. DISTRIBUTION/AVAILABILITY STATEMENT
Approved for public release; distribution is unlimited.

12b. DISTRIBUTION CODE

13. ABSTRACT *(maximum 200 words)*

An off-the-shelf data logger was used as the basis to evolve software and hardware installations providing a simple, reliable data recording system for UAV flight tests. Wiring harnesses, circuit board and plug designs, as well as controlling software were developed for general installations. The recorder is housed in a 4x2.5x1.5 inch box which can be conveniently installed or removed in any UAV. It is capable of storing up to 512K of data at sampling rates up to 3200 Hz with eight, 12-bit analog channels. A set of MATLAB commands was developed to allow convenient processing and analysis of recorded data. Numerous ground and bench tests were conducted as well as flight tests.

14. SUBJECT TERMS
Unmanned Air Vehicle, Flight Test, UAV Data Acquisition

15. NUMBER OF PAGES
144

16. PRICE CODE

17. SECURITY CLASSIFICATION OF REPORT Unclassified	18. SECURITY CLASSIFICATION OF THIS PAGE Unclassified	19. SECURITY CLASSIFICATION OF ABSTRACT Unclassified	20. LIMITATION OF ABSTRACT UL

NSN 7540-01-280-5500

Standard Form 298 (Rev. 2-89)
Prescribed by ANSI Std. 239-18 298-102

Approved for public release; distribution is unlimited

DEVELOPMENT OF ONBOARD DATA ACQUISITION FOR UNMANNED AIR VEHICLE FLIGHT TESTING

Joseph M. Merola
Lieutenant, United States Navy
B.S. Aviation Engineering, Ohio State University, 1988

Submitted in partial fulfillment
of the requirements for the degree of

MASTER OF SCIENCE IN ASTRONAUTICAL ENGINEERING

from the

NAVAL POSTGRADUATE SCHOOL
December 1996

ABSTRACT

An off-the-shelf data logger was used as the basis to evolve software and hardware installations providing a simple, reliable data recording system for UAV flight tests. Wiring harnesses, circuit board and plug designs, as well as controlling software were developed for general installations. The recorder is housed in a 4x2.5x1.5 inch box which can be conveniently installed or removed in any UAV. It is capable of storing up to 512K of data at sampling rates up to 3200 Hz with eight, 12-bit analog channels. A set of MATLAB commands was developed to allow convenient processing and analysis of recorded data. Numerous ground and bench tests were conducted as well as flight tests.

TABLE OF CONTENTS

LIST OF FIGURES

List of Tables

ACKNOWLEDGEMENTS

Thanks to Joachim Beer, a voice of reason through a stressful time and a true friend for all time.

I. INTRODUCTION

A. MISSION NEED

Whether in the minds of science fiction dreamers or on the drawing boards of aircraft developers, pilotless aircraft have been of interest since the beginning of aviation. In the early days, Unmanned Air Vehicles (UAVs) enjoyed small developmental victories. In September 1924, a Curtiss N-9 float-equipped "Jenny" biplane took off from the Potomac River near Dahlgren, Virginia, flew a triangular course with climbs and glides and landed successfully back onto the river: the first flight of a Remotely Piloted Vehicle (RPV)! [Ref. 1:p. 295]

UAV development was slow and mostly relegated to experimental uses until the Vietnam War. Then, a hastily improvised UAV, the Teledyne Ryan's AQM-34 'Lightning Bug', was used with great success [Ref. 1]. Based on the promise the UAV showed during the conflict, aerospace developers pursued a number of advanced designs. However, most ended up shelved after the military and political support never came. A man-in-the-cockpit aircraft justified the existence of many and held greater appeal than UAVs to decision makers of the time. [Ref. 2]

In the 1980's the utility of UAVs was beginning to become obvious. In 1982, Israel conducted a successful campaign in the Bekaa Valley located in southern Lebanon. Much of the success was attributed to the integrated use of UAVs against the Syrian forces. The Israelis used MASTIFF and SCOUT UAVs to relay video pictures as well as

electronic parameters of SAM radar units to airborne command posts. The UAVs provided valuable information and helped minimize Israeli losses. [Ref. 3:p.266-271,277]

In 1985, after spending a week in Israel observing UAV operations, Secretary of the Navy John Lehman directed that a short range UAV be procured using existing technology and off-the-shelf equipment [Ref. 4:p. II-1]. Lehman was "convinced that remotely piloted vehicles or RPVs could identify targets on the ground and spare pilots from danger [Ref. 5:p. 1A]." This resulted in the development and acquisition of the PIONEER UAV from Israeli Aircraft Industries (IAI). The PIONEER was a derivative of the Israeli SCOUT which was used very successfully by the Israeli military. The PIONEER was the first assigned to newly formed Navy and Marine Corps units. Although the PIONEER progress was not without its difficulties, it represented the beginning of an upward trend in UAV development and acquisition. [Ref. 6:p. 4-6]

The upward trend begun in the 1980's has become an explosion of UAV development and employment in the 1990's. This explosion can be attributed to two events in recent history: the collapse of Soviet Communism and the Iraqi invasion of Kuwait [Ref. 2]. During the Gulf War, despite a relatively small number of UAVs deployed, the intelligence gathering and battlefield employment results were dramatic. In one case, a Marine task force commander was able to monitor UAV imagery of a Kuwaiti city as his troops approached, revealing the reaction of Iraqi forces to Marine armor,

artillery and troop movements [Ref. 7]. The Army used UAV reconnaissance data to brief and prepare Apache helicopter pilots prior to missions. UAVs proved useful in searching for mines, assessing battle damage, spotting for gunfire support, and locating key Iraqi targets. Perhaps the most interesting incident is shown in reconnaissance video of Iraqi soldiers attempting to surrender to a UAV. [Ref. 8:p. 86-87]

The unquestionable success of UAVs is not the only reason for the marked increase in UAV technology and employment. The end of the Cold War has created an environment with drastically shrinking defense budgets and is forcing military leaders to rethink the way they conduct business. Now, more than ever, the goal is to get the most war fighting capability for the least amount of money. Compared to manned aircraft, UAVs are simple, inexpensive and easily transported. They can be launched from land, sea or air. They are generally small in size, produce a minimal signature, are more survivable than full size aircraft and, like manned aircraft, are recoverable and reusable. They can operate in high risk, dangerous or sensitive areas without the chance of losing or injuring a pilot. The advantages are numerous and unparalleled increase in UAV interest indicates the world is beginning to take notice.

Numerous countries around the world are finding both military and nonmilitary applications. Military uses include terrain mapping, remote sensing, target search, target detection, classification and identification, reconnaissance and surveillance and delivery of munitions. Promising nonmilitary uses include law enforcement

(counter-narcotics surveillance and border patrol), environmental monitoring (meteorological, atmospheric, agricultural, and hazardous waste sites), emergency response (fire fighting, disaster relief, and search and rescue), and surveying. [Ref. 9:p. 608-609]

Every major country has some sort of UAV development. Countries such as Belgium, Canada, France, Germany, Israel, Italy, Russia, China, and Sweden are only a few actively developing unmanned vehicles. In the Spring of 1994 South Africa used their "Seeker" UAV to carry out day and night monitoring of the widely dispersed polling stations in the country's first free elections. [Ref. 2]

Many universities and small companies are expanding their development in UAV programs. The relatively low cost involved affords itself nicely to university settings which are high in educational motivated research and low in material funding. Young, fresh approaches explored in research settings find their way to commercial and military applications directly by published findings and indirectly through the minds of students moving into the work force. One forecast predicts production of nearly 8,000 recoverable UAVs, valued at almost $4 billion during the decade 1994-2003. This is for air vehicle cost alone, a mere 15% of total system costs. Another survey estimated around $2 billion worth of potential business in civil UAVs alone by 2000. [Ref. 2] Given that testing and data collection expand the science as well as the applications for UAVs, the current unparalleled explosion of interest and research in UAV technology ensure the role of

Unmanned Air Vehicles in the world will continue to expand well beyond anything seen to date.

B. STATEMENT OF OBJECTIVE

An essential part of UAV development is flight test and evaluation. To accomplish this, numerous data systems have been used to characterize flight qualities and record performance. Systems range from complicated telemetry recording systems to simple visual observation of the flight. Each has its advantages and disadvantages and is selected based on the objectives and resources available. The resources available depend on current technology and cost.

Past tests at NPS using onboard analog tape recording or in-house telemetry systems have been only partially successful. A new data acquisition system using a low-cost data logger has been developed in an attempt to provide a robust yet simple means of recording airborne data. The data logger unit, Tattletale Model 5F, is produced by Onset Computer Corporation in Massachusetts. The unit includes an A/D board, is small and inexpensive, and is capable of storing up to 512K of data. The objective of this work was to develop baseline procedures for using the Tattletale 5F to record and analyze UAV flight test data, expanding the knowledge base of data recording at the Naval Postgraduate School and providing a robust, simple, cost-effective system to increase the UAV research capability of the Navy.

II. BACKGROUND

A. PREVIOUSLY AT NPS

At NPS, a number of methods have been used to collect data from UAV flight tests. In September 1990, LT James D. Salmons conducted research on a half-scale Pioneer UAV to estimate the behavior of a full-scale Pioneer. Salmons used a seven-channel TEAC analog recorder to record flight test data. The recorder was carried onboard the UAV during flight and the tape was removed after the flight for processing. A large tape playback unit was used to play the data tape for conversion to a digital format where it could be processed using commercial software and saved on a computer. Because the data recorder was located on the aircraft, it was highly susceptible to aircraft vibration. Despite complex mounting procedures attempting to isolate the recorder from vibration, the recorded data contained large amounts of noise rendering the data virtually unusable. [Ref. 10]

In September 1991, LT Kent R. Aitcheson conducted follow-on research with the half-scale UAV. To correct the vibration problem associated with data recording, Aitcheson used the CHOW-1G telemetry system designed by Kevin T. Wilhelm at NPS. The telemetry system consisted of a seven-channel airborne transmitter and a ground-based receiver. The transmitter multiplexed the analog inputs from the sensors, converted them to digital signals, then sent them to the ground-based receiver. The receiver converted the signal back to analog format for recording by the TEAC tape recorder [Ref.

11:p. 16]. LT Paul Koch also successfully used this system for his research in March 1992 with the half-scale Pioneer UAV. Although the system removed the vibration from the data, it still required the same, cumbersome pre-flight set-up and post-flight processing required by Salmons' system.

Another data collection system was used by LT Eric J Watkiss for his Master's thesis work in March of 1994. Watkiss used a commercially available telemetry system, the Digicon TT-01 Tele Tachometer/ASI, to determine the airspeed. The control surface deflections were measured in an indirect manner by determining the correspondence between the pilot's remote control-stick deflection and the control surface deflection. In flight, the control-stick deflection was visually checked and recorded. Although this system was simpler than the system used by Aitcheson and Kotch, it did not give actual, accurate measurements of control surface deflection.

Accurate data measurements and other major problems associated with the systems of the past can be solved by using a UAV data recording system based on the Tattletale Model 5F. The system developed is simple to install. After spending less than two hours to install the aircraft wiring harness, the data recorder unit can be mounted or removed in minutes. The recorder is capable of accepting data inputs from a wide variety of analog and digital sensors, providing a means for accurate data measurement. In addition, because the recorder relies on digital technology to save the data in RAM, it is not subject to the vibration noise encountered with an analog tape recorder.

Not only is the recorder less complicated and easier to install than previous systems used at NPS, it has greatly simplified the entire process of collecting flight data. After arrival at the field, the Tattletale is ready to go with less than two minutes of preflight checks. During the flight, the pilot controls recording from his remote control console. Post-flight operations require less than five minutes to download the data from the Tattletale's RAM to a file on a laptop computer. After download, the 5F is ready to record another flight. The data file saved on the computer is processed and plotted, ready for analysis with a single MATLAB command.

Development of this flight data recorder makes the data recording aspect of UAV flight testing less cumbersome and expands the test capability of the UAV lab at NPS. It will facilitate further study of UAVs by allowing the engineer to concern himself with UAV aerodynamics and design rather than the mechanics of how to collect the flight data.

III. EQUIPMENT

A. GENERAL DESCRIPTION

The UAV data recording system is designed to be a simple, low-cost method for acquiring flight test data. The system consists of the Onset Computer Corporation Tattletale Model 5F housed in a plastic box which is 4 inches long by 2.5 inches wide by 1.5 inches high. Inside the box, the Tattletale plugs into a circuit board which is wired to a DB-25 female plug in the side of the box. The complete unit weights approximately 4.0 ounces. A wiring harness is prewired at appropriate points inside the aircraft and converges to a male DB-25 plug which matches the plug on the recorder box. The multipin plug allows for easy installation or removal from the aircraft.

The recorder is programmed using a software development system supplied by Onset Corporation called TxTools. TxTools runs on a host IBM PC or Macintosh and works in collaboration with a ROM control program running on the Tattletale to form an interactive BASIC development system. TxTools provides a window programming editor for developing BASIC programs. It also provides a compiler for generating code and a terminal program to provide debugging and interaction with the Model 5F. The ROM control program on the recorder communicates with the computer through a serial port. It is able to accept and execute BASIC programs as well as interact with the user by printing and offloading data for later analysis.

Data is supplied to the recorder through a series of sensors. Up to eight sensors may supply data to the recorder at one time. The sensors must supply an input voltage of 0.0 to 5.0 VDC. If required, a regulated power supply of 5.0 VDC may be supplied from the data recorder to the sensors ensuring the maximum voltage input capacity of the recorder is not exceeded.

Once data is collected, it is downloaded to a file using TxTools. Commands from a MATLAB toolbox called Tattle5F can then be executed to process the data, converting it to an ASCII format. After conversion to an ASCII format, the data may be easily analyzed using commands from the Tattle5F toolbox. The toolbox contains five custom MATLAB commands to process the Tattletale data, preparing it for display and analysis or processing by other means.

B. HARDWARE

1. Tattletale 5F

The Onset Computer Corporation manufactures small, inexpensive data loggers for a wide variety of applications. One of the company data loggers is the Tattletale 5F (Figure 1).

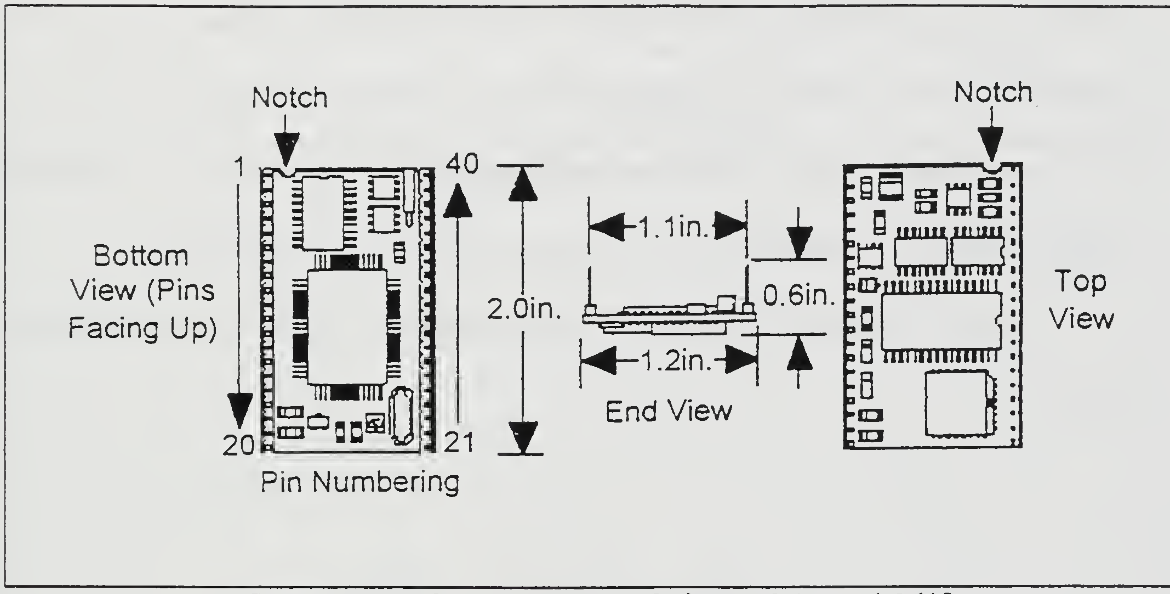


Figure 1 - Tattletale Model 5F [After Ref. 12:p. 6-263]

The 5F is a specialty computer board with integrated hardware and software components which simplify the design of battery-based data loggers. It is an inexpensive, off-the-shelf product which allows the user to forego the problems typically associated with custom microprocessor design projects. Power management, input-output,

communications, timing, software development, and driver design issues have been solved ahead of time.

The Tattletale 5F has two on-board regulators, current and thermally limited for protection from unintentional overloads during development, with low-power management circuitry for direct connection to a 9.0 VDC battery or any 6.0 to 15.0 VDC power supply. The unit has eight analog inputs with a 12-bit A-D converter for direct measurement from sensors and 14 digital inputs and outputs with timing and counting functions. There is an RS-232 communications port, nonvolatile program storage and RAM data storage of up to 512K. The small data loggers have been used for applications such as aerospace, oceanography, racing, agriculture, and medical research.

The 5F is made up of four basic sections displayed in Figure 2 and described in Table 1.

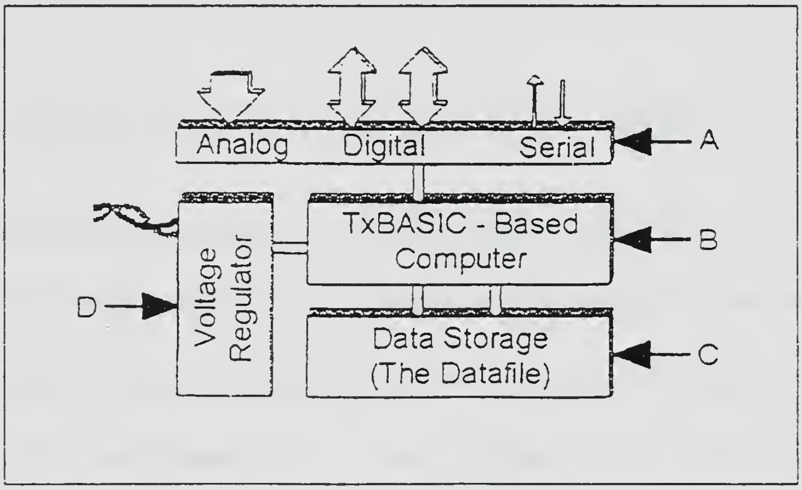


Figure 2 - Four Major Sections of Tattletale [After Ref. 12:p. 1-19]

Diagram Section	Description
A	Analog and digital I/O, including UARTs, individually programmable digital I/O lines, counter, square, wave generator, and three wire serial interface.
B	CMOS CPU. CMOS RAM and FLASH EEPROM for non-volatile program storage.
C	Data storage (the Data file for storing the results of measurements.
D	Voltage regulator to control supply voltages from a battery input or 6-15V power supply.

Table 1 - 5F Section Description [After Ref. 12:p. 1-19]

A general description of the Model 5F's specifications is shown below.

Size (inches)	1.2x2x.5
Weight (ounces)	.5
Processor	6303
Data capacity (RAM)	480K
Prog.capacity (Flash EEPROM)	16K
Analog Channels	eight 12-bit
Maximum sampling rate (Hz)	3200
Analog input voltage (default)	0-5V
Digital I/O lines	14
Minimum Current	3.1mA
Peak Current	30mA
UART baud rate (default)	19200
Power supply range 6 to 15V	Real-time clock
Program language	TxBasic
Operating Temp	0C to 70C

Table 2 - Model 5F Specifications [After Ref. 12:p.1-18]

The Model 5F's components are wired to a 40-pin circuit board which can be plugged directly into an easy to wire prototyping board supplied by the Onset Corporation. The pin functions are shown in Figure 3.

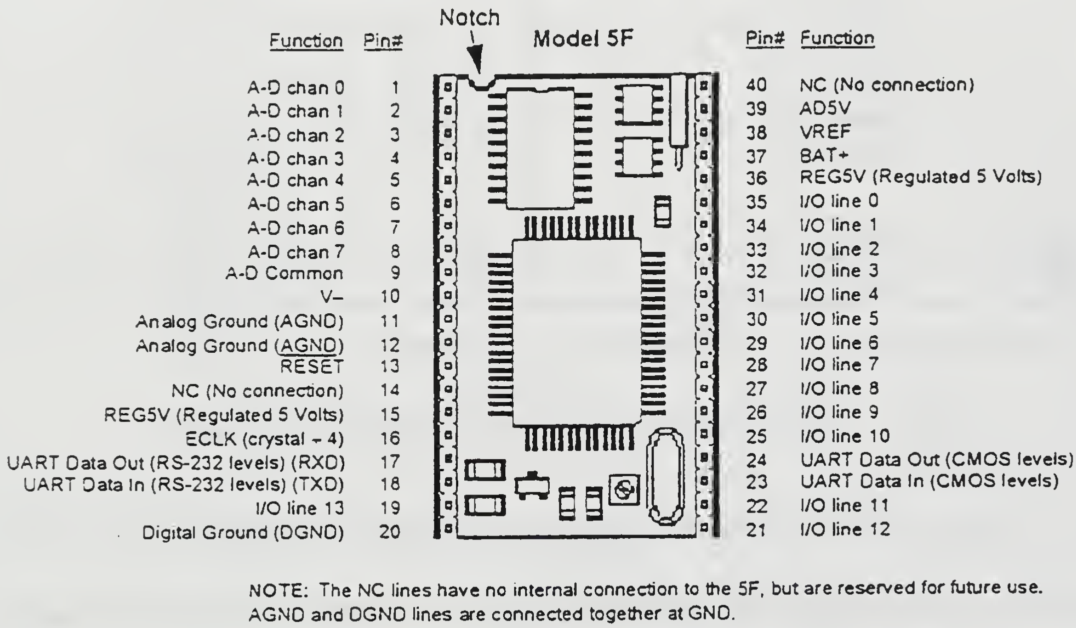


Figure 3 - Model 5F Pin Functions [After Ref. 12:p. 6-264]

2. Recorder Box

The recorder box is a plastic 4x2.5x1.5 inch box available from the local electronics supply store. The box contains a printed circuit board, prototyping board, and a 40-pin receiver plug for mounting the Tattletale circuit board. The prototype board is wired to a DB-25 female plug which protrudes out one end of the box (see Figure 4). The wiring between the prototyping board and the 25-pin plug is shown in Table 3.

Circuit Board Pin #	Description	DB-25 Connector Pin #	Circuit Board Pin #	Description	DB-25 Connector Pin #
1	Channel (5)	1	9	Sensors Common Ground (NEG)	9
2	Channel (1)	2	17	Serial Data Out	17
3	Channel (2)	3	18	Serial Data In	18
4	Channel (3)	4	20*	Input Ground (NEG)	20
5	Channel (4)	5	35*	I/O Pin (0)	28
6	Channel (5)	6	36	Reg. 5.0 VDC Output	24
7	Channel (6)	7	37	Input Power (POS)	25
8	Channel (7)	8			

* NOTE: On Circuit board, I/O Pin (0) wired to Input Ground via 10 K 1% resistor.

Table 3 - Prototype Board Pin-to-Plug Wiring Guide.

Only 15 of the 25 available pins were used on the DB-25 pin connector. This allows for later continued development of the recorder system. Although no specific applications are currently being considered, there are ten pins available on the plug which might be used for more advanced digital sensor input and output taking full advantage of the Model 5F's capabilities.

3. Aircraft Wiring Harness

A standard wiring harness has been developed for installation in the aircraft being tested. With the aircraft wiring harness in place, installation and removal of the data recorder is convenient. A diagram of the wiring harness is shown in Figure 4.

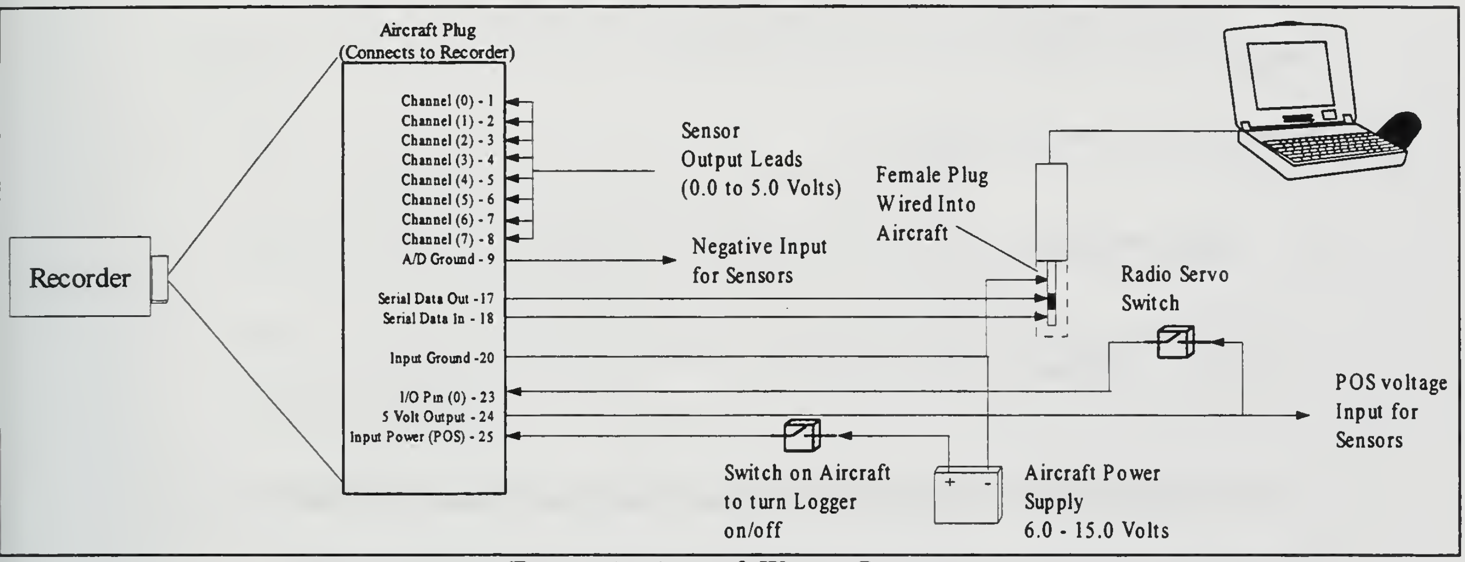


Figure 4 - Aircraft Wiring Diagram

Two switches are wired into the aircraft. The first is a standard two-position toggle which turns power to the recorder on and off. The second is a switch which can be toggled from the pilot's remote control console to control the start and stop of recording. A three-lead, 1/4-inch female plug is wired into the aircraft allowing serial interface with a computer for loading and unloading data. The power supply may come from any 6.0 to 15.0 VDC source. Sensor output leads are wired into the first eight pins of the DB-25

connector. The outputs from the sensors must not exceed 5.0 VDC or damage could occur to the recorder.

Three wiring harnesses were prewired. The first was used to connect to the recorder for bench testing. The second was wired into the Naval Postgraduate School's FOG-R UAV for flight tests. The third harness was wired into the LoFlyte UAV developed by LT Michael Fendley to record flight test data. The details and results of flight tests are covered in Chapter IV.

C. SOFTWARE

1. TxTools

TxTools is software provided by the Onset Computer Corporation which provides an interactive development system using a Tattletale BASIC derivative, TxBasic. TxTools creates a terminal environment for interacting directly with the Tattletale. It also provides a text editor for loading and creating TxBasic programs for the recorder. The ROM control on the Tattletale communicates with the computer through the serial port accepting and executing BASIC programs, as well as providing an interface for offloading logged data for analysis.

2. Data Recorder TxBasic Program

TxBasic is the operating system used to control the functions of the Tattletale Model 5F and has minor differences from standard BASIC. The differences are not significant and involve some predefined terms and standards which make it easier to program the Tattletale. All of the commands and procedures are entered through TxTools.

The program written accomplishes the following tasks:

- Start and stop of recording is controlled by the UAV pilot.
- With initial power application, there is a positive check for proper power supply to the recorder.
- With initial power application, there is a check for proper operation of the pilot's ability to control the recording.
- Recording can be restarted at any time.
- Data can be offloaded at the completion of the flight.

The program is executed when the data recorder is switched on. Table 4 shows an explanation of the actual TxBasic program loaded into the EEPROM of the Model 5F.

model 510	This tells TxTools which model of Tattletale is being programmed. Some commands are model specific.
dfPoint = 0	dfPoint is the index for keeping track of where the data is stored in the Tattletale. This commands resets it to zero to ensure data begins being stored at the beginning of the memory stack.
onerr fulmem	This line assures there will be no error when the recorder is allowed to run beyond the maximum recording time. When the memory is full, the program jumps in the program to put the recorder in standby mode waiting to unload the data. • onerr command causes the program to jump to the designated location if there is any error in program execution. Here it is used to indicate when the memory is full.
print print "UAV Data Recorder" print "Version 24 NOV 1996" print print print "*****************************" print print "Tattletale is receiving power." print "Program is running properly." print print "Toggle remote control switch to RECORD." print	These messages are displayed on the computer screen when the recorder is first turned on. This gives a positive check for the user; the recorder is receiving power.

Table 4 - TxTools Program Explanation

init: if pin(0) > 0 goto onmsg goto init	Loop which allows the tattletale to wait until the remote-control RECORD switch is toggled to on. Gives a chance to ensure computer is hooked up to recorder prior to continuing on with software checks. • This is a programming loop which looks for I/O Pin (0) of the Tattletale to receive a 5.0 VDC signal. The input of the 5.0 VDC signal is controlled by the remote control radio switch. When 5.0 VDC is sent to I/O Pin (0), the program continues with its execution.
onmsg: print print "************************************" print print "Recorder radio switch toggled on properly." print print "Toggle remote control switch to OFF." print "************************************"	This messages printed on the screen after initial power application and after the remote control switch is toggled to RECORD. This gives a positive check to the user that the remote control switch toggles on properly.
bgnprg: if pin(0) > 0 goto bgnprg	Allows recorder to wait until remote-control is toggled to OFF before setting up to record data. • Loop similar to the one above. As long as the 5.0 VDC is supplied to I/O Pin (0), the program remains in this holding loop.

Table 4 - TxTools Program Explanation (continued)

print print "Recorder radio switch toggled off properly." print print "CHECKS COMPLETE" print "Recording will begin when the transmitter switch" print "is toggled to RECORD...." print	This message is displayed on the computer screen and indicates the remote control record switch toggled to OFF properly. This also indicates the end of the initial checks. The recorder is now ready to record data.
wait: if pin(0) > 0 goto rcrd cbreak finish goto wait	Tells recorder to wait until remote-control is toggled to RECORD before continuing on to record data. After recording is finished, program stays in this loop until remote control toggled back to RECORD or [ctrl]+[c] received from computer. "cbreak" line tells recorder to jump to the "finish" label if it receives a [cont]+[c] command from the computer. Allows capability to find ending address of recorded data when ready to offload it to a file.
rcrd: dfPoint = 0	"rcrd" is a label indicating the start of the program commands which control recording. Data file Pointer. dfPoint is a variable which contains the address of the location where the first data point is stored. It is set to zero to ensure data is saved at the beginning of the data storage area.
cnt = 0	Variable used to keep track of the number of data "bursts" taken. A burst is a set of data points taken at the same time.

Table 4 - TxTools Program Explanation (continued)

rate 2	Used in conjunction with the following two sleep commands to control the interval between samples. Refer to Appendix F, Changing Sampling Rate and the Tattletale Operation Manual Section 5, "Rate" command for details.
sleep 0	Resets the internal clock to ensure timing is correct for the data points.
while pin(0) > 0	Begins a 'While' loop which tells the recorder to continue to record points as long as the remote control is toggled to RECORD. • As long as I/O Pin (0) is receiving 5.0 VDC, the recorder will record data.
sleep 5	Makes the recorder wait for 5*1/200 of a second between successive sleep commands. Makes the recorder sample at a rate of 40 Hz. Works in conjunction with "rate" command above to obtain the 40 Hz. • This is the command which must be modified along with the rate command to change the sampling rate of the recorder. See Appendix F, Changing Sampling Rate for more details
burst dfPoint,8,2	Beginning at the address stored in dfPoint, "8" channels of data are recorded at the same time. Each channel of data uses "2" bytes of memory. i.e. one data point is recorded for each channel all at the same time.

Table 4 - TxTools Program Explanation (continued)

cnt = cnt + 1	Increases the counter in order to keep track of the number of bursts of data, i.e. the number of data points taken for each channel.
if cnt%40=0 print cnt/40," "	This is a counter which counts the seconds the recorder is toggled to RECORD. If a computer is connected, it displays the seconds count on the screen. This line prints a count in seconds on the computer screen if connected. This allows for troubleshooting and bench testing of the recorder.
wend	Ends the While loop for recording data points.
goto wait	Sends the program back to a loop to allow the recorder to wait to record data again or download data.
fulmem: if pin<0> > 0 goto fulmem goto wait	This small loop puts the recorder in standby if the memory gets filled up and the remote control record switch is still togged to RECORD.
finish: print print "End of Data Pointer = ", dfPoint-1	Subroutine which is run when computer is connected and [ctrl]+[c] is pressed. Prints address of end of recorded data. This address is used later to offload recorded data to file.

Table 4 - TxTools Program Explanation (continued)

3. MATLAB

TxTools is used to offload data from the data recorder and save it as a binary file. To allow for convenient display and analysis of the data, a file saved in ASCII format

was desired. This would allow the data to be manipulated using common software such as Microsoft Excel or Lotus.

Initially, a shareware software program, Display Tattletale Data, posted on the Internet site of Onset Computer Corporation, was used to display the data and convert it to an ASCII format. After conversion, the data was graphed using Excel. However, this process was cumbersome. Although posted on the Onset Internet site, the company did not support the software and there was no documentation to explain operation. A phone call to Nova Computer Systems of Nashville, Tennessee, the reported authors of the program, resulted in puzzled responses with affirmations their company made no such software. Trial and error resulted in a series of procedures to operate the software.

To make the data recorder more user friendly, MATLAB was chosen for development of a series of commands which could convert the data to an ASCII format and display it in a useful manner. This made use of the poorly-documented Display Tattletale Data software, and the slow spreadsheet programs unnecessary. A MATLAB toolbox defining five MATLAB functions was developed. The toolbox, Tattle5F, contains six files, one for each function and one which describes the contents of the toolbox making "help" available for the recorder functions while running MATLAB.

A description of each of the files and its function is listed below. Table 5 shows a list of file types which may be generated. The MATLAB code for each function is in Appendix A.

File Type	Description
filename.dat	Raw data file. Tattletale binary format.
filename.bin	Raw data file. Intel binary format.
filename.txt	Raw data file. ASCII format.
filename.red	Reduced data file. ASCII format.

Table 5 - File Extensions.

a) dat2bin.m - Data to Binary

This function converts a file from Tattletale binary format to Intel binary format. The data from the data recorder is originally saved in a format with two-byte words for each data sample: the first byte is the most significant byte and the second byte is the least significant byte. Intel binary format has the least significant byte first and the most significant byte second. The function, dat2bin, opens the data file, reads the bytes, swaps them and then saves them in another file of the same name with a ".bin" extension. The original file is left intact.

b) bin2asc.m - Binary to ASCII

The bin2asc function generates an ASCII format file from an Intel binary formatted file. The command reads the binary file and resaves it in an ASCII format to a file of the same name with a ".txt" extension.

c) redasc.m - Reduce ASCII

This MATLAB function reads a raw data file in ASCII format, converts the data to its appropriate units, and saves it in another ASCII format file. For example, the raw data recorded by the recorder for the elevator deflection will be in units based on the voltage measured by the elevator sensor. The redasc command converts the units to degrees of deflection based on calibration values.

To do the conversion, sensor calibrations are conducted and data recorded. The coefficients from the conversion equation are put into a matrix used as input to the command. Each of the eight channels will have a conversion equation making the matrix 8x3 in size. The command reads a channel of raw data from the file, converts it using the appropriate row of coefficients, and saves it to a new file. The process is repeated until all eight channels have been reduced. The new file containing the reduced data has the same file name as the original file with a ".red" file extension.

The details of the conversion can be seen in the MATLAB listing, redasc.m, shown in Appendix A: MATLAB Programs. Samples of data calibration data and the calibration equations are shown in Appendix E: Calibration Data.

d) plotasc.m - Plot ASCII

This MATLAB function plots an ASCII data file. It may be used to plot the -.txt raw data files or the -.red reduced data files. Up to six channels may be plotted on the same chart. Initially, the full range of the plot is shown. To facilitate analysis of

the data, the user may reduce the range of the plot by using the mouse and a crosshair presented on the screen. This allows the user to look at very small portions of the data.

e) redplot.m - Reduce and Plot

Redplot.m converts a Tattletale data file and plots the data. The other functions defined in the Tattle5F Toolbox are used to provide a one-step means of looking at the data. Besides plotting the data, this function generates data files with -.bin, -.txt, and -.red extensions for later use.

f) contents.m - Tattle5F Help File

This file is contains a listing of the MATLAB files in the Tattle5F Toolbox. It provides help to the user by showing appropriate helpful information regarding the Tattle5F Toolbox. When "help Tattle5F" is typed, the information in this file prints on the screen.

g) senscals.m - Sensor Calibrations

Senscals.m is a MATLAB list file which contains the sensor calibration coefficients used to reduce raw data. It merely defines an 8x3 matrix variable which contains the coefficients. Prior to reducing data, this simple program may be run to define the calibration matrix variable. It precludes typing the entire matrix in every time a user wishes to reduce data. When the sensor calibration coefficients change, this file must be changed manually using a text editor. The file is in the toolbox as a convenience.

The calibration matrix variable may be defined by other means as well. Appendix F, Changing Calibration Coefficients, contains information about changing this file.

IV. OPERATIONAL TESTS

A. INTRODUCTION

Following development of the software and hardware configurations, operational tests were conducted. The operational tests fall into two categories: bench/ground checks and flight tests. All of the testing involved numerous iterations beginning with investigation of capabilities and desired applications, software development, hardware development/installation, and evaluation of performance. The following sections describe the equipment as well as the ground and flight tests performed.

B. EQUIPMENT DESCRIPTION

1. FOG-R

a) Physical Description

The FOG-R UAV was born from the FOG-M UAV program which was a Fiber Optics Guided UAV to deliver missiles on target. The "-R" model of the FOG was designed to provide reconnaissance information about "enemy" targets and the "battlefield" environment [Ref. 13]. The airframe was acquired by NPS when the program was discontinued. At NPS, the aircraft is used for research and because of its

payload capabilities and availability, provided an ideal test platform for the UAV data recorder.

The FOG-R UAV is a high wing, single-engine aircraft. Its general characteristics are shown below and a line drawing is shown in Figure 5. [Ref. 14:p. 3]

Wing Span:	122 inches
Wing Chord:	20 inches
Engine Type:	Two-Cycle, Air-Cooled
Engine Power:	12 Horsepower
V_{max}:	122 MPH
Take-off gross Weight:	92 pounds

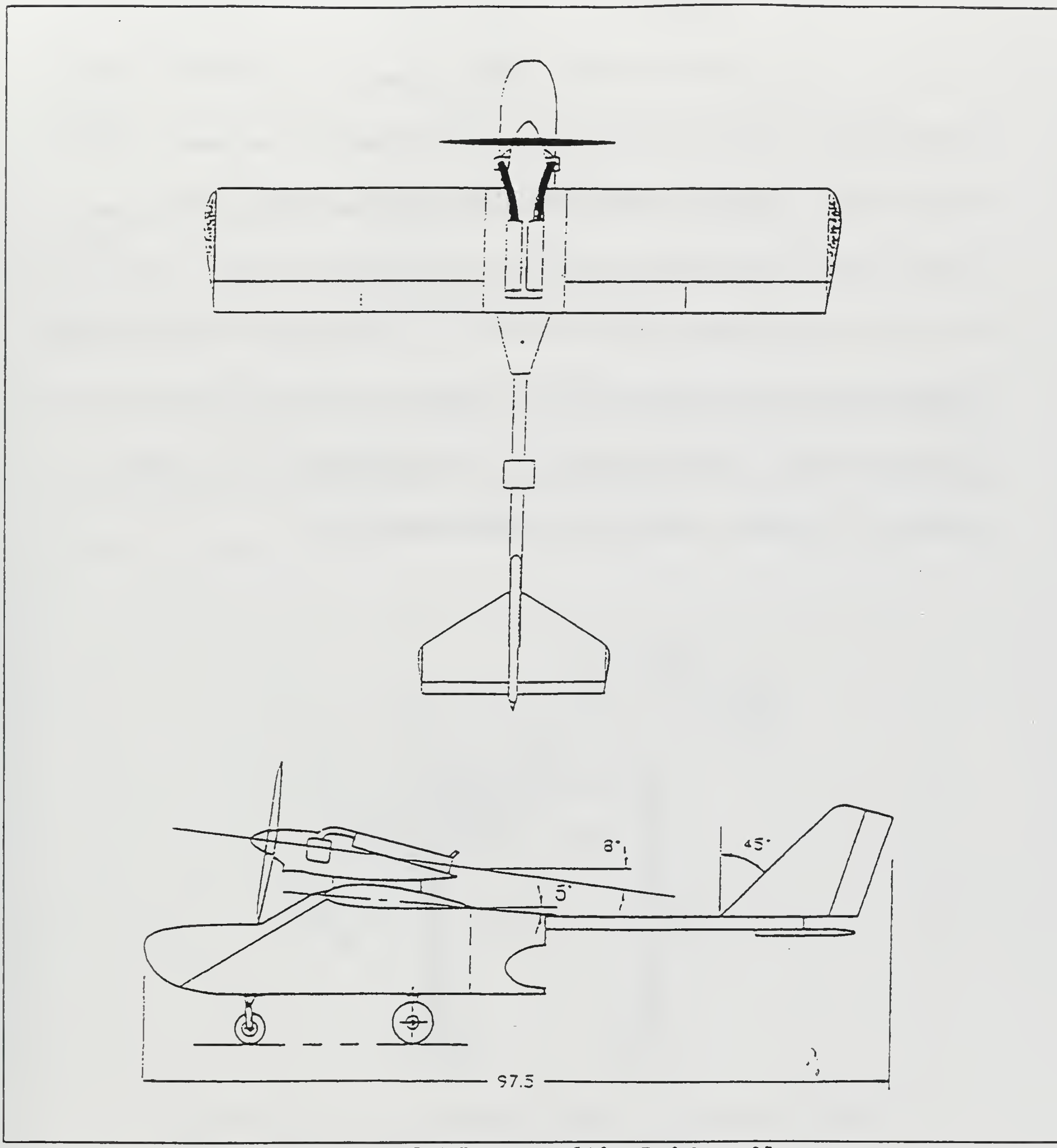


Figure 5 - FOG-R Drawings [After Ref. 14:p. 2]

b) Sensors

The FOG-R was equipped with six sensors. Four of the sensors were potentiometers from the New England Instrument Company and were used to measure aileron, elevator, rudder and alpha deflection The potentiometers were MKV rotary position, low torque potentiometers with a resistance tolerance to ±5.0%. The remaining two sensors were pressure sensors from Sensym, Inc. and were used to measure dynamic and barometric pressure giving airspeed and altitude data. All sensors received power from the regulated 5.0 VDC power supply of the data recorder. Table 6 contains a summary of the sensors used to measure the flight parameters.

Parameter	Sensor type	
Aileron Deflection	Potentiometer	New England Instruments, Co. MKV rotary type
Elevator Deflection	Potentiometer	
Rudder Deflection	Potentiometer	
Alpha	Potentiometer	
Altitude	Sensym, Inc ASCX15AN 0-15 psia absolute pressure sensor	
Airspeed	Sensym, Inc. 0 to 1 psi differential pressure sensor	

Table 6 - FOG-R Sensor Types

The pressure sensors, shown in Figure 6, have a maximum input voltage of 20 VDC with an increasing positive output voltage directly proportional to the absolute or differential pressure.

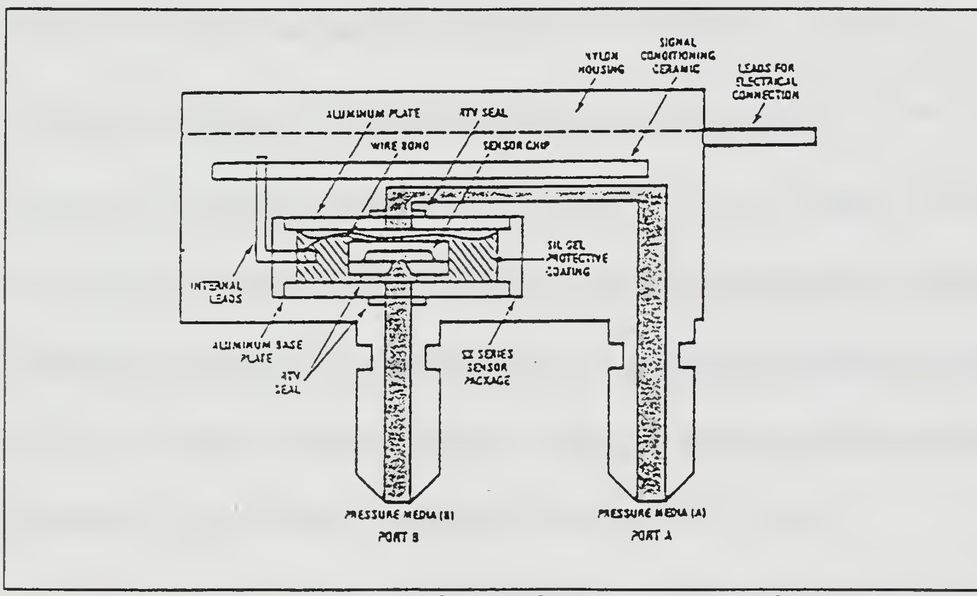


Figure 6 - Sensym Pressure Sensor Cutaway [After Ref. 15:p. 2-9]

The sensors are ratiometric to the supply voltage causing proportional changes in the offset voltage and full-scale span with changes in the supply voltage. Figure 7 shows a diagram of the electrical connections for the pressure sensors.

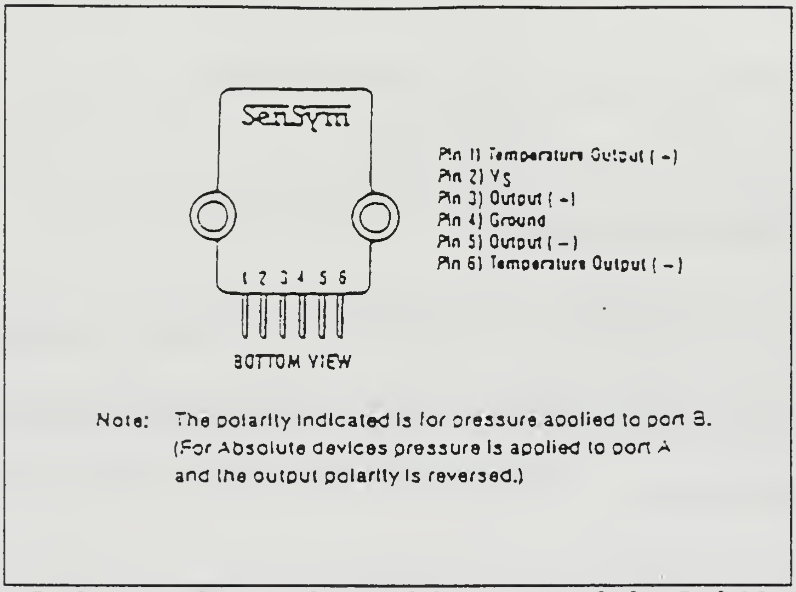


Figure 7 - Pressure Sensor Electrical Connections [After Ref. 15:p. 2-3]

For the absolute sensor used for the altitude, port B is inactive and pressure is applied to port A. The absolute sensor has a sealed vacuum reference chamber. This makes the offset voltage defined by the vacuum pressure. Since all pressure is measured relative to a vacuum reference, all changes in barometric pressure or changes in altitude will cause changes in the sensor output voltage.

Both sensors were surrounded by 3 inch foam and mounted in the nose of the aircraft. Rubber tubing connected the pressure ports to a pitot-static probe protruding from the nose of the aircraft.

To measure control deflection, each of the control surfaces had a potentiometer mounted near the surface, a bellcrank mounted on the surface and a rod connecting the two.

Any deflection of the surface caused a corresponding resistance change of the potentiometer. Alpha (angle of attack) was measured by attaching a vane directly to a potentiometer mounted to a pitot-static probe at the front of the aircraft. The input voltage for the potentiometers was the regulated 5.0 VDC supplied by the data recorder. The output leads of the potentiometers and the pressure sensors were wired to a central terminal strip to allow easy switching and controlling of sensor to data recorder channel inputs. The potentiometers were "centered" to make a midrange deflection correspond to approximately 2.5V output. Calibrations were performed to obtain the deflection-to-voltage output correspondence. The results of the calibration are shown in Appendix E.

c) Recorder Installation

Prior to installation of the recorder itself, the aircraft prewired harness was constructed following the wiring diagram in Figure 4. The harness was installed in the aircraft allowing the recorder to be mounted in the center of the fuselage. Two terminal strips were mounted in the middle of the fuselage.

The first terminal strip bridged the sensor outputs to the eight data recorder input channels. The second terminal strip supplied power to the various sensors and the data recorder. Using the terminal strips allowed easy changing of sensors

configurations and convenient control over which data recorder channels received sensor inputs. Power for the recorder came from a dedicated 6.0 VDC, 3.0 Ah gel-cell battery. The power supply was regulated to 12.0 VDC using a DC to DC converter.

The recorder itself was wrapped in 1/4-inch foam and fastened using hook-and-loop closures to a shelf in the middle of the fuselage. The male plug of the aircraft wiring harness was routed to the data recorder position. In this position, the data recorder was easily removed or installed as required.

The on/off switch for the recorder and the plug for the computer interface were mounted at the rear of the fuselage on the right side of the aircraft. This position allowed easy interaction with the data recorder in the field without requiring any aircraft disassembly.

2. LoFlyte

a) Physical Description

The LoFlyte UAV is an unpowered, scaled-down version of the LoFlyte UAV developed by Accurate Automation Corporation with a grant from the National Aeronautics and Space Administration (NASA). Both UAVs are being used to research aspects of a Mach 6 Wave rider design. The NPS LoFlyte UAV was built by LT Michael Fendley as a thesis research project (see Figure 8).

Fendley's LoFlyte UAV is 75 inches long with a 44-inch wingspan. It is constructed with light plywood, Styrofoam, and fiberglass. A UAV data recorder was installed in Fendley's model and flown to investigate the low-speed glide characteristics of the LoFlyte design as well as the operational aspects of the UAV data recorder.

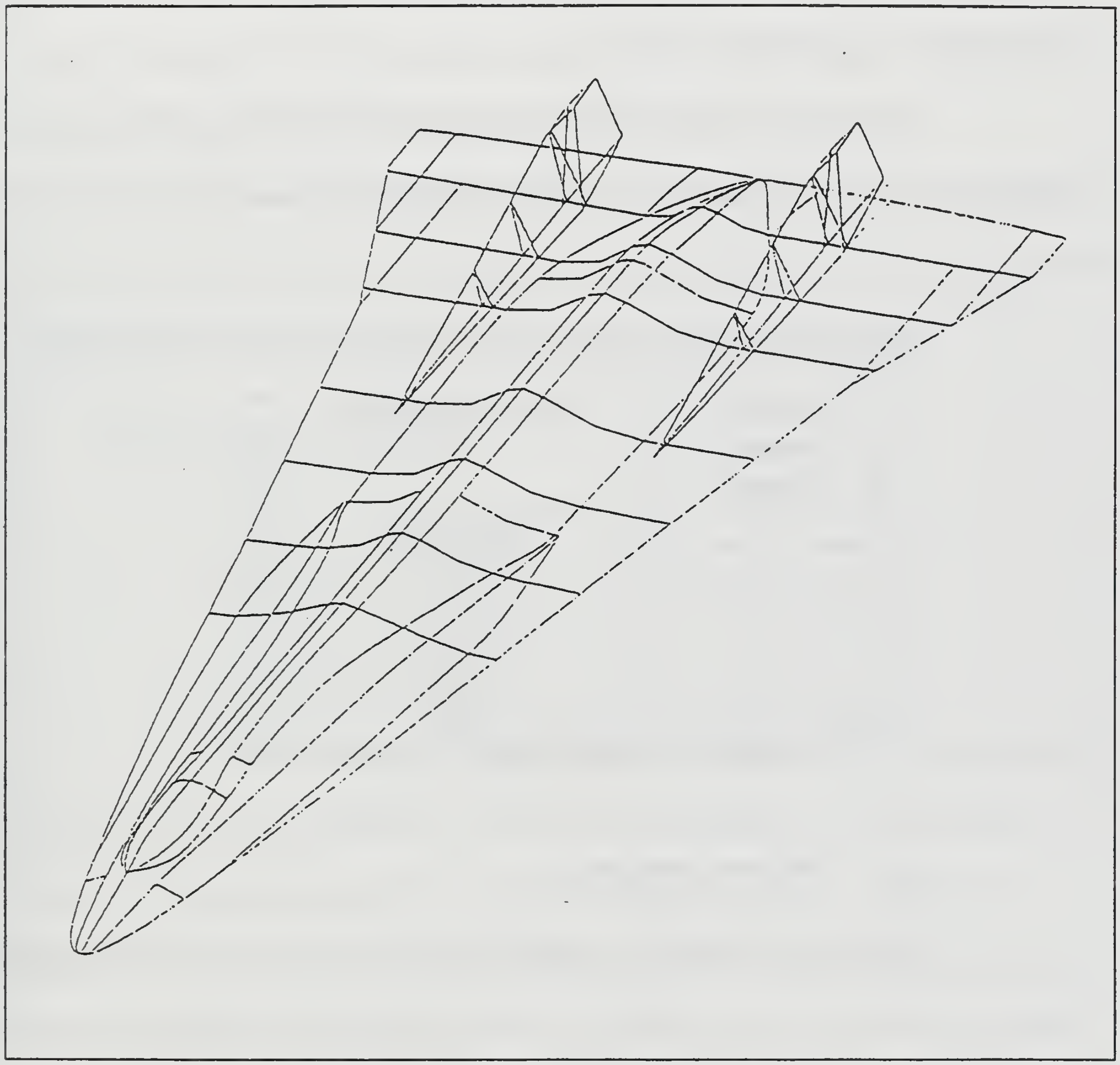


Figure 8 - LoFlyte Drawing [After Ref. 16]

b) Sensors

The LoFlyte UAV was equipped with six sensors of the same type used in the FOG-R. The pressure sensors were mounted in the nose of the aircraft. Rubber tubing connected the pressure ports to a pitot-static probe protruding from the nose of the aircraft similar to the FOG-R installation.

To measure control deflection, the control surfaces had potentiometers mounted near the surfaces, bellcranks mounted on the surfaces and a rod connecting the two.

Table 7 describes the sensor output to data recorder input wiring scheme.

Data Recorder Channel	**Sensor Description**
0	Right Elevon
1	Left Elevon
2	Rudder
3	Airspeed
4	Altitude
5	Alpha

Table 7 - LoFlyte Channel Allocation

c) Recorder Installation

Prior to installation of the recorder itself, the aircraft prewired harness was constructed following the wiring diagram in Figure 4 with the recorder placed in the nose of the aircraft. The harness was connected directly to each of the sensors and the recorder

power supply. The power was supplied by a 10.8 VDC, 500 mAh, nine-cell NiCad battery.

Because the standard recorder case was too large for installation in LoFlyte, the recorder was disassembled and a smaller, custom case was constructed to house the recorder. The LoFlyte recorder case was 0.75x2.0x3.0 inches. It was secured in the nose of the LoFlyte UAV and plugged into the aircraft wiring harness.

The on/off switch for the recorder and the plug for the computer interface were mounted on the top front of the fuselage. This position allowed easy interaction with the data recorder in the field without requiring any aircraft disassembly.

C. BENCH/GROUND CHECKS

1. Bread Board

a) Description

For initial testing and development of the UAV data recorder software, the Tattletale Model 5F was mounted on an electronic bread board. To simulate sensor inputs, a 10K thermistor and a 10K 0.1% resistor were connected to Channel (0) (see Figure 9). This allowed room temperature measurements to be used as input data for the recorder during software development.

As the capabilities of the Model 5F were explored, the requirements of the UAV data recorder were defined. The result was a bread board setup which simulated an aircraft wiring harness. A Radio Shack 9.0 VDC power supply, catalog number 273-1552, was used to power the bread board.

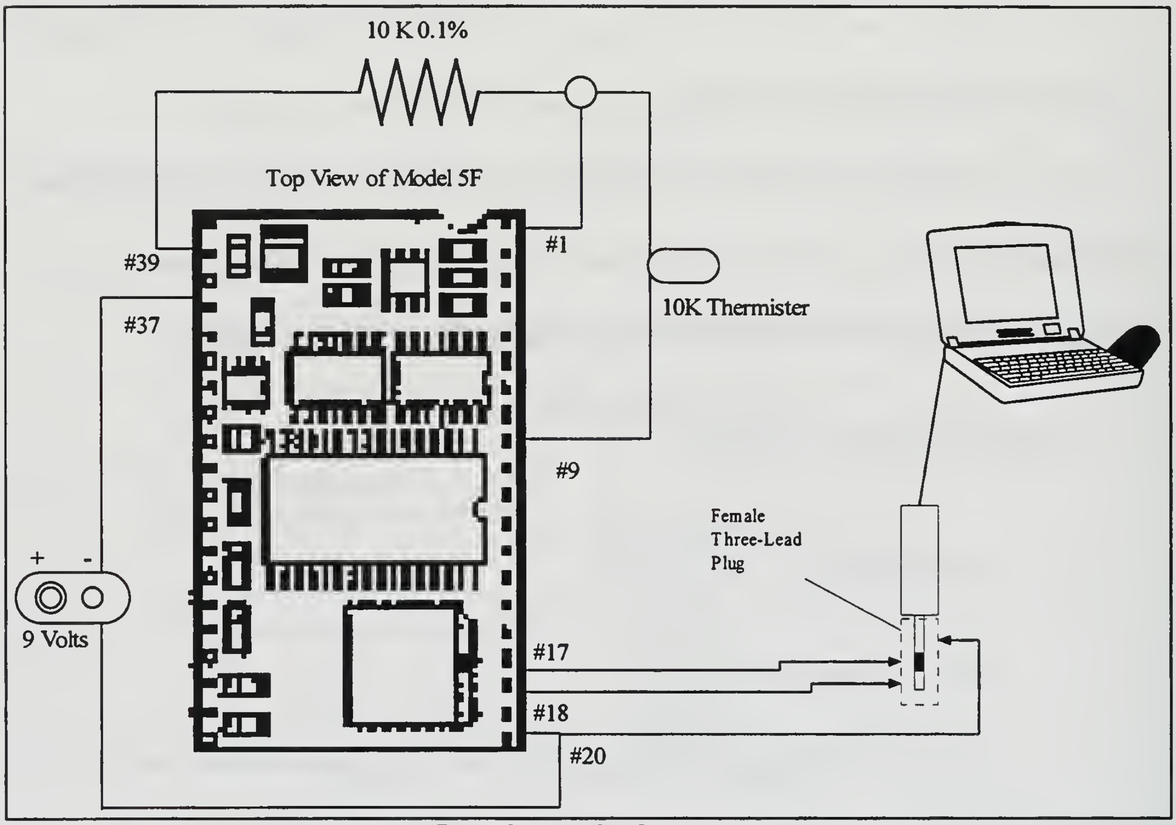


Figure 9 - Bread Board, Thermistor Diagram

A series of TxBasic programs were written, slowly expanding the capability of the recorder until the desired performance was achieved. Input/output channel (0) is the key to the operation of the recorder. This channel controls the start and stop of recording. Whenever a “high” signal, 2.0 to 5.0 VDC, is received at I/O channel

(0), the recorder software allows the recorder to record inputs from input channels zero to seven.

Initially, the 5.0 VDC supply from the Model 5F was connected directly to I/O channel (0) via a switch. However, bench testing with the bread board setup showed the recorder randomly began recording whenever a voltage was applied to one of the input channels. Further investigation revealed the I/O pin voltages fluctuated unless positively set "high" or "low." Although the I/O channel was positively set "high", when the 5.0 VDC was removed, the voltage seen at the pin fluctuated due to internal connections of the Tattletale. This caused the recorder to initiate recording at random times when the 5.0 VDC was not applied.

To solve this problem, a 10K 1% resistor was wired between the Model 5F pin 35, I/O channel (0), and pin 20, input ground. This provided a positive "low" signal, 0.0 VDC, to the input/output channel. When the switch controlling the 5.0 VDC "high" signal was closed, I/O channel (0) was "high." Thus, a positive "high" or "low" signal was provided to ensure control over the start and stop of data recording.

After finalizing the hardware configuration, the bread board configuration was used to refine the software programming of the Model 5F. By adding a DC to DC converter with a 12.0 VDC output, the Tattletale was supplied the 12.0±0.6 VDC required to modify its EEPROM. By modifying the EEPROM, the recorder was able to run the TxBasic program immediately upon power-up of the data recorder. This enabled

the final modification and programming of the Tattletale Model 5F prior to installation in the aircraft.

b) Remarks

The bread board configuration provided a convenient means to develop the hardware and software. The drawback was its confusing layout of connections and wiring. Although appropriate for initial development it is not practical as a long term development tool for the data recorder. Because of this, a “bench” box should be developed to facilitate later development and testing of the data recorder. This will provide a means for later expansion and improvement of the data recorder without the cumbersome bread board “spaghetti” wiring.

2. Sensor Calibrations

a) Description

Calibration of the sensors was required to get a correspondence between the displacement of the measured parameter and the voltage output of the sensor. The actual calibration data for both FOG-R and LoFlyte are shown Appendix E. The calibrations involved measuring the displacement of a parameter and using a multimeter to measure the voltage output of the sensor. After obtaining measurements at several points, throughout a parameter range, the data were plotted and a line was fit to the data.

The equation of the line was later used to determine displacement throughout a parameter range.

Aileron, elevator, rudder, and alpha used potentiometers as sensors. To measure the actual deflection of each, a simple protractor was placed at the pivot point of the respective control surface. Multimeter leads were placed at the ground and output of the potentiometer being measured. With a 5.0 VDC input, the output of the potentiometers ranged from 0.0 to 5.0 VDC. The sensors were adjusted to be "centered" at approximately 2.5 VDC output with 0.0 degrees of deflection. Although voltage output ranges varied with control deflections, resolution was approximately 1 degree, limited more by the method of deflection measurement than the accuracy of the potentiometer.

After the potentiometers were "centered", the control surfaces were moved through incremental angles until maximum deflection in both directions was obtained. For each increment, the corresponding voltage output was read from the multimeter and recorded. Linear, second-order , and third-order line fits were generated from the data and compared. In all cases, the linear line fit was sufficient to describe the relationship between control deflection and voltage output.

The altitude sensor was calibrated by using a calibration manometer. This provided a means to apply a known pressure to the Sensym 0.0 to 15.0 psi absolute pressure sensor. With a 5.0 VDC input, sensor output ranged from approximately 0.25 to

4.75 VDC over the 15.0 psi range of the sensor. This provided an ideal resolution of approximately 7.0 feet.

A barometer was used to record the ambient pressure at the time of calibration. The ambient pressure was added to the calibrator pressure for each measurement. This gave a true measure of the absolute pressure during the calibration. The manometer was used to apply incremental pressures to the sensor. At each increment, the corresponding output voltage was measured and recorded. The data were plotted and the equation for the line fit was used in the standard -atmosphere equation below.

$$H = \frac{1 - \left(\frac{P_a}{P_0}\right)^{\frac{1}{5.2561}}}{6.87535 10^{-6}}$$

H = Height (ft)

Pa = Ambient Pressure (lb/ft^2)

Po = Standard Sea Level Pressure (lb/ft^2)

This allowed a function for the voltage-to-altitude correspondence to be generated. A linear curve fit was sufficient to describe the correspondence and was used for the data reduction.

Airspeed calibration data were acquired using the calibration manometer also. Known pressures were able to be applied to port A of the 0.0 to 1.0 psi differential pressure sensor. With a 5.0 VDC input, sensor output ranged from approximately 0.25 to 4.75 VDC over the 1.0 psi range of the sensor. This provided an ideal resolution of approximately 0.5 ft/sec.

Incremental pressures were applied to the sensor and the corresponding voltage output was recorded. This data were plotted and an equation for a line fit was generated. The equation coefficients are the items used by the MATLAB functions to reduce the raw data.

b) Remarks

The quality of the calibrations directly affect the outcome of the data. The more accurate the calibrations are, the more accurate the data will be. A problem with the technique for calibrating the control surfaces is the manual measurement of the deflections. Although index lines were drawn on the control surfaces and used to read the deflections from a simple protractor, the position and stability of the protractor presented a large factor of variability. Also, the perspective of the person reading the protractor presented variability.

Accuracy was approximately 1.5 degrees with the given method. This could be improved by taking many more data points and averaging the results. However, a more reliable improvement would be to develop a means of securing the protractor to the surface. This would provide consistent placement of the measuring device with respect to the surface being measured resulting in more accurate measurements of deflection.

Accurate calibration of the pressure sensing devices was contingent on the calibration manometer. If it varied the pressure applied to the sensors, the voltage output

varied and was not reproducible. To prevent this from happening, it was important to ensure there were no air leaks in the manometer or any of the connective tubing. Without leaks, reliable, consistent readings were obtained and used.

3. Bench Checks

a) Description

Tests were conducted in the modeling lab initially with the FOG-R and later with LoFlyte. After installation of the aircraft wiring harness, the data recorder was installed in the aircraft. With the aircraft configured for flight, a series of tests were conducted to ensure the data recorder operated as desired. More than 12 bench tests were conducted. Not only were the tests a means of checking the correct operation of the hardware and software but also provided valuable guidance in establishing operating procedures for the data recorder.

With the aircraft configured for flight, a laptop computer running TxTools was connected. The UAV pilot operated the recorder from his remote control console while the recording was monitored at the computer. UAV controls were cycled at various increments for various times. Measurements were made using a protractor at the control surfaces. The measurements were later compared to the data plots to verify the validity of the data and the plots.

From the beginning, all of the hardware operated as expected. Figure 10 is a sample of data acquired during one of the tests.

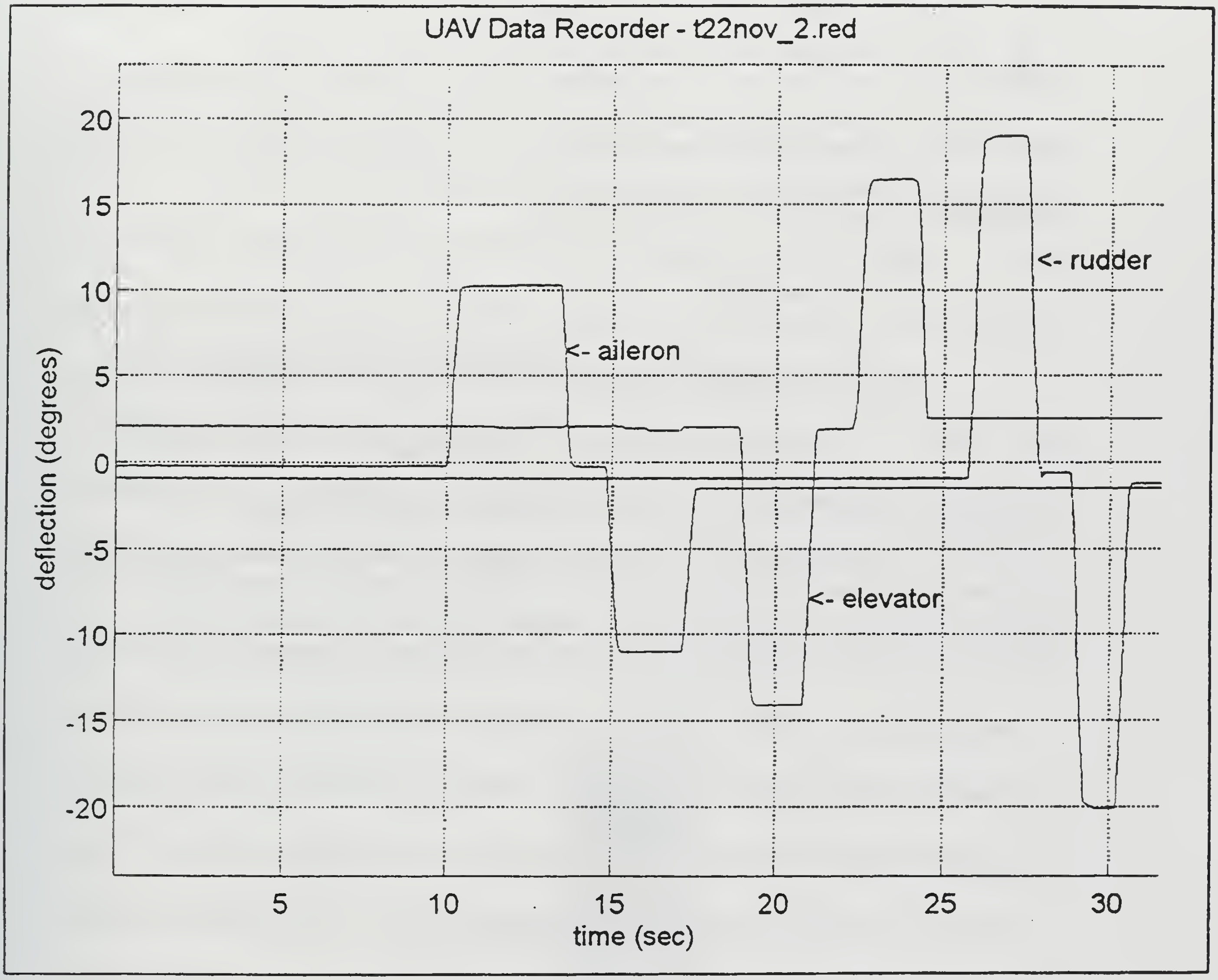


Figure 10 - Sample Recorder Data

The data in Figure 10 show the deflection of the ailerons, elevator, and rudder. As can be seen, the elevator seems to have approximately 2.0 degrees down bias.

Although it is possible the elevator is maintaining the 2.0 degrees down, it is more likely the calibration was slightly inaccurate. In this case, it was necessary to inspect the elevator, check the manual deflection measurements taken, and recheck the calibrations to establish confidence in the data. Investigation revealed the elevator was maintaining 1.0 degree down and the calibration was slightly off. Corrections were made to the calibration coefficients and the data were reduced properly.

b) Remarks

Initially, the hardware worked well; however, after approximately three tests, it began to start and stop recording randomly when a control surface was moved. The wiring was checked and it was determined the electronic switch used to toggle the recorder was malfunctioning. The switch was replaced by a more reliable manual switch toggled by a standard RC servo. Figure 11 shows a top view of the servo switch.

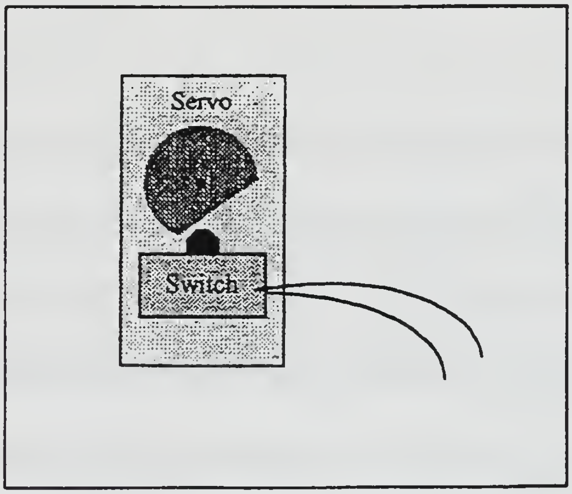


Figure 11 - Radio Servo Switch

This switch proved very reliable throughout the remainder of testing and no further hardware malfunctions were experienced.

Only two TxBasic software malfunctions arose during the testing. The first occurred when tests were done which recorded beyond the recorder's memory capacity. The data recorder attempted to save data in a memory location which did not exist. This caused a run-time error and forced the Tattletale to cease operation. To solve this, some lines of code were added causing the recorder go into standby if the memory filled up. Procedures to unload the data are the same and full memory has no consequence to the user except that no more data may be recorded.

The second software malfunction also was discovered while conducting a test to fill the memory of the data recorder. The address of the last data point obtained by the recorder is displayed on the computer screen when [CTRL]+[C] is pressed. However, initially, this address actually was the address of the *next available* memory location. The result was an error when the memory was full and the user attempted to offload the data. The address given as the last data point was one more than the last data point and the memory address did not exist. Therefore, an error occurred when the user told the computer to unload data up through that data point. A simple correction to the TxBasic code solved the problem and no further TxBasic software errors occurred.

Because of thorough development and testing of the data recorder configuration on the bread board, the problems with hardware and TxBasic software were

minimal. However, the bench tests were very valuable in the development of the operational procedures and MATLAB software for the data recorder. The repetition helped ensure recording operations went smoothly in the field.

4. Ground Runs

a) Description

Ground runs consisted of configuring the FOG-R for flight, starting the motor and recording data. This provided a vibrating environment similar to the environment the data logger would operate in during flight. For most of the tests, the aircraft was secured to the ground preventing it from moving while a laptop computer running TxTools was connected. The engine was started and the UAV pilot operated the recorder from his remote control console while recording was monitored at the computer. The UAV controls were cycled at various increments for various times with the engine running at idle, half throttle and full throttle. Later, the data were analyzed to investigate the effects of vibration on the recorder and the sensors. Two of the ground runs were conducted with the computer disconnected and the aircraft taxiing. A total of approximately 14 ground runs were conducted.

b) Remarks

Initially, the recorder appeared to experience no effects from the vibration. However, on the second ground run, the recorder started and stopped recording

intermittently. The problem indicated an interruption in the 5.0 VDC “high” signal going to input/output channel (0). Two items were identified as possible causes for the interruptions. The first was a faulty switch. The second was bad wiring or connections in the lines from the 5.0 VDC power supply to I/O channel (0). The switch was checked and cycled numerous times and no malfunctions could be found. The wiring was checked and the terminal screws connecting the wiring were tightened. Also, sponge padding was placed along the sides of the terminal blocks reducing the vibration of the connective wiring. No further malfunctions were experienced. The conclusion was the malfunction had been caused by a loose wire.

The more important item revealed by the ground runs was the effect of vibration on the sensors. The potentiometers were not effected by the vibration. However, the pressure sensors were significantly effected. Because the range of pressures covered by the Sensym pressure sensors was large, they would be impractical for precise characterization of the flight characteristics of a UAV.

Although the aircraft was not moving, the noise caused by vibration resulted in indications of as much as 27 ft/sec. To reduce the error, the pressure sensors were mounted in several different orientations and a ground run was conducted with each new orientation. Analysis of the airspeed plots indicated the least amount of noise was obtained with the pressure sensors packed in 3-inch foam in the nose of the aircraft, with pressure ports sticking up, oriented from front to back.

Although the error caused by the noise remains significant, sensors with a smaller pressure range should work well. The error caused by the noise would be acceptable. Figure 12 is a chart showing the dynamic pressure to airspeed conversion. For the anticipated operating speeds of the FOG-R, a differential pressure sensor of 0.0 to 5.0 inches of H_2O would be best suited for airspeed measurements. Based on the noise experienced by the 0.0 to 1.0 psi differential pressure sensor and the resolution of a 0.0 to 5.0 inches of H_2O pressure sensor, anticipated resolution of the latter is approximately 0.5 ft/sec. This could be improved with a pressure sensor which covers a smaller range. However, the compromise to less noise error is a reduction in the measurable operating airspeeds of the aircraft.

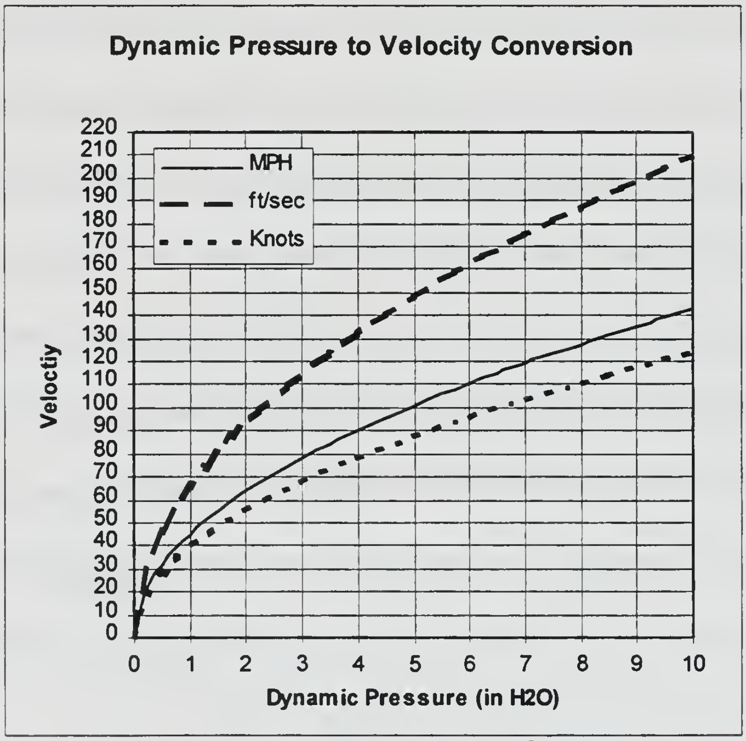


Figure 12 - Dynamic Pressure to Velocity

The altitude resolution could be improved, as well, by use of an absolute pressure sensor with a smaller range. The range of the sensor used in the FOG-R was 0.0 to 15.0 psia. This resulted in an ideal resolution of approximately 7.0 feet. With the vibration noise, this was reduced to approximately 20.0 feet. A sensor with a range of 12.0 to 16.0 psia would give an ideal resolution of approximately 1.5 feet and a resolution of approximately 4.0 feet with vibration noise. As with the airspeed, a better resolution could be obtained by using a sensor with a smaller pressure range. However, the measurable operating altitude would be decreased. To obtain better resolution for altitude measurements, another type of sensor would have to be used.

D. FLIGHT CHECKS

1. FOG-R

The FOG-R was unable to be flight tested due to weather and scheduling conflicts. However, follow-on research is expected to involve flight testing of the FOG-R UAV. Because the data recorder was ground tested in a flight configuration under conditions similar to flight and, in some cases, more severe, no problems are anticipated.

2. LoFlyte

a) Description

Two flights were conducted with the UAV data recorder installed in Fendley's LoFlyte UAV. The first was conducted in Monterey, California at Monterey Beach. The second was conducted by NASA Dryden Flight Research Center at Edwards Air Force Base.

For the first flight, the data recorder was switched on and the UAV was thrown by hand from a sloping hill at Monterey Beach. The flight lasted approximately five seconds. The aircraft landed hard on its nose, severely damaging the first 18 inches of the nose. No damage occurred to the data recorder. However, the impact tore the battery pack from the aircraft, causing a loss of the flight data stored in the data recorder.

The UAV was repaired and modified for the second flight by NASA. The aircraft was equipped with a parachute which could be deployed in an emergency to prevent damage. The battery pack was mounted securely to prevent a recurrence of the loss of power to the data recorder. Fendley's UAV was attached to the underside of a NASA "mothership" UAV. The mothership was flown to altitude and the LoFlyte UAV was released. The flight lasted for approximately 38 seconds. Positive control of the aircraft was not achieved and the aircraft's safety chute failed to deploy. The aircraft impacted the ground at approximately 14.0 ft/sec. No data were able to be offloaded.

The impact of the aircraft resulted in a momentary interruption of the power supply to the data recorder. This caused the recorder to reset and lose the data stored.

The impact of the aircraft shoved the nose skid through the fuselage directly into the area the data recorder was mounted. The top access hatch to the data recorder area was damaged and shifted. The hatch may have jolted the recorder on/off switch which protrude through a hole in the hatch. This would have caused a momentary interruption of the power.

b) Remarks

In both flight tests of the LoFlyte, data collection was not successful. However, in both cases, the data recorder was exposed to conditions which it was not originally designed for. The recorder was not designed to be an aircraft "black box" for accident investigation. Nevertheless, the recorder could be slightly modified to better withstand such situations. A back-up power supply could be built into the recorder to prevent any loss of damage in high impact situations. Details of how the recorder might be modified are discussed in the recommendations section.

All other aspects of the data recorder worked well. It performed well and proved simple to operate.

V. SUMMARY AND RECOMMENDATIONS

A. CONCLUSIONS

The UAV data recorder performed well in all aspects and has the potential to perform well in a variety of UAV testing applications. The wiring required for installation is minimal and straight forward. The simple, small recorder box permits easy, non-intrusive installation with convenient removal when required. In the field, preflight preparation requires merely a notebook computer and only a few seconds of system checks.

Post-flight download of the data is simple and takes only a few minutes. The MATLAB Tattle5F toolbox offers a means to reduce and view the data at the airfield, providing quick feedback on the quality of the data. The toolbox commands are simple and require only a basic knowledge of MATLAB.

The recorder can accept inputs from a variety of sensor inputs with the only limiting factor being the voltage input range, 0.0 to 5.0 VDC. However, this is a common voltage range for a variety of sensors and generally poses no obstacle to data collection.

Its simple design makes the UAV data recorder an attractive option for collection of flight data. The attraction is its simplicity and reliability. Although it does not offer the "bells and whistles" of a more complex real-time design, it is ideal for collection, characterization and analysis of UAV flight data. It fills a need for such a system and

enhances the UAV research capability of both the Naval Postgraduate School and the United States Navy.

B. RECOMMENDATIONS

Although the data recorder performs well, the potential of the Tattletale Model 5F is much greater. A number of things could be done to expand the capability of the data recorder. One of the easiest and most useful improvements is a backup power supply for the data recorder. For applications where there is expected exposure to harsh conditions, such as was with the LoFlyte UAV, a backup power supply could ensure the flight data would not be lost due to interruption in primary power. A 9.0 VDC power supply could be wired in parallel with the normal power supply to pins 37 and 20 of the Model 5F. The 9.0 VDC battery could be installed inside the data recorder case when needed. Power to the recorder would be maintained at all times.

Without a backup power supply, the integrity of the aircraft wiring harness should be carefully checked and the quality and placement of the recorder power switch should be carefully considered to ensure no interruptions in power.

For increased accuracy, different sensors could be explored. The sensors used for the test installations used analog sensors. The computer age has made available a host of digital sensors which would be less prone to electronic and vibration noise than conventional analog sensors.

The susceptibility of pressure sensors to vibration noise could be rendered inconsequential with the use of other types of sensors for airspeed and altitude measurements. For airspeed, a small propeller might be capable of supplying data. For altitude, a radar altimeter may be adaptable for use in a UAV.

Also, the ease with which the recorder software can be changed lends itself well to changing the recorder operations. In its current configuration, the recorder records continuously. It is possible to modify the software to provide a means for placing electronic markers in the data to signal the start or end of a flight maneuver. The electronic marker could be keyed using the pilots remote control record switch or, with hardware modifications, a second switch could be used to mark the data.

Finally, the recorder was chosen and constructed to be convenient and expandable. The possible improvements are limited mainly by the imagination of the user. The most important recommendation is to not accept conventional limits. The UAV data recorder is simple yet powerful and can be used for many applications.

APPENDIX A: MATLAB Programs

A. FUNCTION BINARY TO ASCII (BIN2ASC.M)

```
function bin2asc(filename)

%
% Naval Postgraduate School
% Merola 5 NOV 1996
%
% BINARY TO ASCII - Converts a binary file to an ASCII file.
%
% BIN2ASC('filename')
%
% Input:     filename - (OPTIONAL) - Intel format binary data file.
%            Filename must contain complete path
%            and file extension.
%            Format of file must have
%            word length 2 bytes.  First byte is
%            most significant.  Second byte
%            is least significant. If no file name specified,
%            a prompt allows browsing for file.
%
% Output:    filename.txt - binary file saved in ASCII format
%            with '.txt' extension.
%

if nargin==0
  [filename,path]=uigetfile('*.bin','Select Binary File to Convert to Ascii Format');
else path='';
end

if ~filename
  fclose('all')
  return
end

fname=[path filename];

fid=fopen(fname);

for j=1:length(fname)
  if fname(j)=='.'
    sname=fname(1:j-1);
```

```
break
  end
end

sname=[sname '.txt'];

if nargin==0
  [filename,path]=uiputfile(sname,'Save As (Ascii File Name)');
else
  path=' '];
  filename=sname;
end

if ~filename
  fclose('all')
  return
end

fname=[path filename];

fid2=fopen(fname,'w+');

fseek(fid,0,'eof');
eof=ftell(fid);
fseek(fid,0,'bof');

numpr=0;

while numpr<eof
  a=fread(fid,[8,25000],'ushort');
  fprintf(fid2,'%6g %6g %6g %6g %6g %6g %6g %6g\n',a);
  numpr=2*size(a,1)*size(a,2)+numpr;
end;

fclose('all');

disp('Done');
```

B. FUNCTION DATA TO BINARY (DAT2BIN.M)

```
function dat2bin(filename)

% Naval Postgraduate School
% Merola 10 NOV 1996
%
% DATA TO BINARY - Converts Tattletale data file to file saved in Intel Binary.
%
% BIN2ASC('filename')
%
% Input:     filename - (OPTIONAL) - name of Tattletale file.
%            Filename must contain complete path and file
%            extension.  data will be saved in Intel Binary format
%            which has word length of 16 bytes.  First byte is
%            most significant.  Second byte is least significant.
%            if no filename specified, a prompt allows browsing
%            for file.
%
% Output:    filename.bin - Data file in Intel binary format with
%            '.bin' extension.
%
if nargin==0
 [filename,path]=uigetfile('*.dat','Select Tattletale Data File to Convert to Intel Binary
Format');
else path=‘’];
end

if ~filename
 return
end

fname=[path filename];

fid=fopen(fname,'r+');

fseek(fid,0,'eof');
eof=ftell(fid);
fseek(fid,0,'bof');

for j=1:length(fname)
 if fname(j)=='.'
  sname=[fname(1:j-1) '.bin'];
  break
```

```
  end
end

sfid=fopen(sname,'w+');

numpr=0;

while numpr<eof
  a=fread(fid,25000,'uchar');

  for j=1:2:size(a,1)-1
    least=a(j+1);
    a(j+1)=a(j);
    a(j)=least;
  end;

  numpr=length(a)+numpr;
  %fseek(fid,-length(a),'cof');
  fwrite(sfid,a,'uchar');
end;

fclose('all');
```

C. FUNCTION PLOT ASCII (PLOTASC.M)

```
function plotasc(redfile,RATE)

% Naval Postgraduate School
% Merola 08 DEC 1996
%
% PLOT ASCII - Plots an ASCII data file
%
% PLOTASC('redfile',RATE)
%
% Input:     redfile - (OPTIONAL) - name of ASCII file to plot
%            If no redfile specified, a prompt allows browsing
%            for a file.
%
%            RATE - (OPTIONAL) - the sampling rate in Hz of the
%            data. Default is 40Hz.
%
%            Channel(s) - After command is executed, prompt
%            on screen will ask which channels to plot. Max
%            is 6 channels on one graph. If multiple channels,
%            enter them as an array, e.g. [1 3 5]. Plot colors
%            follow [red green blue cyan magenta yellow].
%
% Output:    A graph of the entire data file.
%
%            The cursor becomes a crosshair. When happy with
%            the ranges of the graph, click the right mouse
%            button on the graph.
%
%            To change the ranges of the graph, set the
%            crosshair on the top left corner of a box surrounding
%            the desired range and click the left mouse button.
%            Next, set the crosshair on the bottom right of a
%            imaginary box surrounding the range desired. Click
%            the left mouse button. Graph will be replotted.
%

if nargin==0
  [filename,path]=uigetfile('*.*','Select Ascii File to Plot');
  rate=40;
else
  path='';
  filename=redfile;
```

```
end

if ~filename
  return
end

if nargin > 1
  rate=RATE;
end

eval(['load ' path filename]);

for j=1:length(filename)
  if filename(j)=='.'
    dat=filename(1:j-1);
    break
  end
end

channels=input('type in which channels to plot eg [1 2 4] ')+1;

numch=size(channels,2);
if numch > 6
  error('Cannot plot more than 6 channels on one chart');
elseif numch<1
  error('Cannot plot zero channels');
end

maxch=size(eval(dat),2);

for j=1:numch
  if channels(j) > maxch
    error('Data does not exist for channels requested.');
  elseif channels(j) < 1
    error('Data does not exist for channels requested.');
  end
end

t=(1/rate):(1/rate):size(eval(dat),1)*(1/rate);

hold on;
grid on;

title('Time Plot of Data');
xlabel('Time (secs)');
```

```
lintyp=[
  'r'
  'g'
  'b'
  'c'
  'm'
  'y'];

for j=1:numch
 ch=channels(j);
 lin=lintyp(j);
 y=eval([dat '(:,ch)']);
 plot(t,y,lin);
end;

button=1;

[x(1),y(1),button]=ginput(1);

while button == 1
  [x(2),y(2),button]=ginput(1);
  if button ~= 1
    break
  end

xmin=x(1);
xmax=x(2);
ymin=y(1);
ymax=y(2);

if xmax < xmin
  tempx = xmin;
  xmin = xmax;
  xmax = tempx;
end

if ymax < ymin
  tempy = ymin;
  ymin = ymax;
  ymax = tempy;
end

axis([xmin xmax ymin ymax]);
```

```
[x(1),y(1),button]=ginput(1);

end;
```

D. FUNCTION REDUCE ASCII (REDASC.M)

```
function redasc(filename,a,rate)

%
% Naval Postgraduate School
% Merola 08 DEC 1996
%
% REDUCE ASCII - reduces an ASCII file using sensor calibration
%              coefficients.
%
% REDASC('filename',a,rate)
%
% Input:        filename - ASCII file which contains the data to
%               reduce using the coefficients supplied.
%
%               a - a 8x3 matrix containing the 2nd order coefficients
%               for each channel of the recorder i.e.
%                       [a11 a12 a13;
%                        a21 a22 a23;
%                        a31 a32 a33;
%                        a41 a42 a43;
%                        a51 a52 a53;
%                        a61 a62 a63;
%                        a71 a72 a73;
%                        a81 a82 a83]
%               Each data point in the ASCII file will be
%               used to calculate a reduced data point by
%               being multiplied through the appropriate
%               2nd order equation with the coefficients
%               shown above.  The first row of coefficients
%               comes from the recorder channel (0) calibration data
%               The second row of coefficients comes from
%               the channel (1) recorder data. etc.
%
%               NOTE: a row of coefficients which has the first
%                       coefficient=999, tells the program this
%                       channel has airspeed numbers which
%                       require special processing.  The coefficients
%                       for airspeed come from the first order
%                       pressure equation derived from the sensor
%                       calibration data.
%
%               rate - (OPTIONAL) - the sampling rate in Hz
```

```
%               of the data.  Default is 40Hz.
%
% Output:       filename.red - reduced data ASCII file with '.red'
%               file extension.
%

if nargin < 2
  error('Sorry, Need to put in filename and Calibration #s with this command');
end
path='';

fname=[path filename];

fid=fopen(fname);

for j=1:length(fname)
  if fname(j)=='.'
    sname=fname(1:j-1);
    break
  end
end

fname=[sname '.red'];

fid2=fopen(fname,'w+');

if nargin < 3
  rate=40;
end

numch=8;

fseek(fid,0,'eof');
eof=ftell(fid);
fseek(fid,0,'bof');
done=0;

while done==0;
  dat=fscanf(fid,'%6g %6g %6g %6g %6g %6g %6g %6g\n',[8,8000]);
  dat=dat';
  dat=(dat./16).*5./4095;
  for i=1:numch
    if a(i,1)==999
      dat(:,i)=( (a(i,2).*dat(:,i)+a(i,3) ).* (2/.0023769) ).^(.5);
    else
```

```
      dat(:,i)=a(i,1).*dat(:,i).^2+a(i,2).*dat(:,i)+a(i,3);
    end
  end
  if size(dat,1)<8000
    done=1;
  end;
  dat=dat';
  fprintf(fid2,'%6.3f %6.3f %6.3f %6.3f %6.3f %6.3f %6.3f %6.3f\n',dat);
  end
end

fclose('all');
```

E. FUNCTION REDUCE AND PLOT (REDPLOT.M)

```
function redplot(datfile,a,rate)

%
% Naval Postgraduate School
% Merola 08 DEC 1996
%
% REDUCE AND PLOT - converts Tattletale data file and plots results.
%
% REDPLOT('datfile',a,rate)
%
% Input:        datfile - Tattletale data file name.
%
%               a - a 8x3 matrix containing the 2nd order coefficients
%               for each channel of the recorder i.e.
%                       [a11 a12 a13;
%                        a21 a22 a23;
%                        a31 a32 a33;
%                        a41 a42 a43;
%                        a51 a52 a53;
%                        a61 a62 a63;
%                        a71 a72 a73;
%                        a81 a82 a83]
%               Each data point in the ASCII file will be
%               used to calculate a reduced data point by
%               being multiplied through the appropriate
%               2nd order equation with the coefficients
%               shown above.  The first row of coefficients
%               comes from the recorder channel (0) calibration data
%               The second row of coefficients comes from
%               the channel (1) recorder data. etc.
%
%               NOTE: a row of coefficients which has the first
%                       coefficient=999, tells the program this
%                       channel has airspeed numbers which
%                       require special processing.  The coefficients
%                       for airspeed come from the first order
%                       pressure equation derived from the sensor
%                       calibration data.
%
%               rate - (OPTIONAL) - the sampling rate in Hz
%               of the data.  Default is 40Hz.
%
```

```
% Output:     datfile.bin - Data file in Intel binary format with
%             '.bin' extension.
%
%             datfile.txt - binary file saved in ASCII format
%             with '.txt' extension.
%
%             datfile.red - reduced data ASCII file with '.red'
%             file extension.
%
%             graph - of the entire data file.
%
%             The cursor becomes a crosshair.  When happy with
%             the ranges of the graph        click the right mouse
%             button on the graph.
%
%             To change the ranges of the graph, set the
%             crosshair on the top left corner of a box surrounding
%             the desired range and click the left mouse button.
%             Next, set the crosshair on the bottom right of a
%             imaginary box surrounding the range desired.  Click
%             the left mouse button.  Graph will be replotted.
%
% for more information, see the following functions:
%       DAT2BIN.M, BIN2ASC.M, REDASC.M, PLOTASC.M
%

if nargin < 2
  error('Sorry, Need to put in filename and Calibration #s with this command');
end

if nargin < 3
  rate=40;
end

dat2bin(datfile);

for j=1:length(datfile)
  if datfile(j)=='.'
    binfile=[datfile(1:j-1) '.bin'];
    ascfile=[datfile(1:j-1) '.txt'];
    redfile=[datfile(1:j-1) '.red'];
    break
  end
end
```

```
bin2asc(binfile);

redasc(ascfile,a,rate);

plotasc(redfile,rate)
```

APPENDIX B: UAV DATA RECORDER CHECKLIST

UAV DATA RECORDER CHECKLISTS

Pre Mission Day

- ☐ Sensors/Pots Properly Calibrated
- ☐ Computer Loaded with Proper Software
    - A. TxTools
    - B. MATLAB
        1. Tattle5F Toolbox
        2. Sensor Calibration Coefficients
- ☐ Bench Test of Data Recorder and All Sensors/Pots
- ☐ Aircraft Batteries Fully Charged

Equipment Checklist

- ☐ **Data Recorder**
- ☐ **Computer**
- ☐ **Data Disks**
- ☐ **Computer** - Recorder Patch Chord
- ☐ **Stopwatch** - to keep track of when in sequence maneuvers happen.
- ☐ **Scales & Blocks** - to do weight and balance checks after fuel burn.
- ☐ **Generator** - to power computer.
- ☐ **Barometer** - to check local barometric pressure.
- ☐ **Thermometer** - to check local temperature.
- ☐ **Protractor/Calibration Tool** - in case calibration is required in field.
- ☐ **Compass** - to determine direction of winds.
- ☐ **Clipboard** - to record data conveniently.

Pre Flight

- ☐ Aircraft Preflighted
- ☐ Data Logger - OFF
- ☐ Remote Control Record Switch - OFF
- ☐ Patch Chord - CONNECT

  Plug the chord into the computer and the aircraft.
- ☐ Computer - START TxTools
- ☐ Data Logger - ON

  Should get start up screen from the data logger. If the Remote Control Record Switch was turned on prior to plugging in or starting TxTools, will NOT damage anything. However, will not get Recorder Initialize Screen.

- ☐ Computer - CONFIRM Recorder Initialize Screen
- ☐ Remote Control Record Switch - RECORD
- ☐ Computer - CONFIRM Recorder Toggle-On Message
- ☐ Remote Control Record Switch - OFF
- ☐ Computer - CONFIRM Recorder Toggle-Off Message
- ☐ Patch Chord - DISCONNECT

  Disconnect chord from aircraft. The Recorder is ready to start recording data.

When Ready to Start Recording

- ☐ Remote Control Record Switch - RECORD

In Flight

- ☐ Time - MONITOR

  At a sample rate of 40 Hz, there is 12.8 minutes of recording time. After 12.8 minutes, the recorder will suspend recording.

When Finished Recording

- ☐ Remote Control Record Switch - OFF

CAUTION: If the Remote Control Record Switch is toggled to RECORD *prior* to offloading data from recorder, it will reset and begin recording. *All previously recorded data will be lost.*

Post Landing

CAUTION: Do NOT secure power to the data recorder before downloading data. All data will be lost if the recorder is turned off.

- ☐ Computer - START TxTools
- ☐ Patch Chord - CONNECT
- ☐ Computer - [CTRL]+[C]
- ☐ Computer - DATA ADDRESS

  Listed on the screen is the End of Data (EOD) address for the data stored in the recorder. This number is required for the data offload. Write it down for later use.

- ☐ Computer - OFFLOAD DATA

  In TxTools, follow menu selections below:

  1. >Tattletale > Offload data file...
  2. Start Address = 0
     End Address = [EOD address]
  3. >Off-load
  4. Type file name with ".dat" extension. >OK
  5. Progress bar will show status.
     Unsuccessful Offload - Error message will pop up. Begin offload procedures again with step (1) above.

     Successful Offload - Progress bar will disappear and control will return to the TxTools Terminal Window.

To Prepare for another flight

- ☐ Computer - [CTRL]+[B]
- ☐ Computer - CONFIRM Recorder Initialize Screen
- ☐ Remote Control Record Switch - RECORD
- ☐ Computer - CONFIRM Recorder Toggle-On Message
- ☐ Remote Control Record Switch - OFF
- ☐ Computer - CONFIRM Recorder Toggle-Off Message
- ☐ Patch Chord - DISCONNECT

  Disconnect chord from aircraft. The Recorder is ready to start recording data.

When Ready to Start Recording

- ☐ Remote Control Record Switch - RECORD

If Last Flight of the Day

- ☐ Patch Chord - DISCONNECT
- ☐ Data Logger - OFF

APPENDIX C: FLIGHT TEST CARD

Flight Test Card

Date:____________

Test Title: Recorder/FOG-R Longitudinal Static Stability Investigation

Objectives:

1. Test Operation of Recorder in flight conditions
2. Establish field operations procedures for recorder
3. Verify data conversion procedures and assess ease of operation
4. Collect data to determine longitudinal static stability characteristics of FOG-R UAV
5. Increase UAV pilot flight experience with FOG-R

General Description:

At a given CG position, fly racetrack pattern with one leg at constant airspeed in trimmed condition. Increase airspeed and fly constant airspeed/trimmed leg again. Want to repeat approximately 5 times to collect trim data for total of 6 different airspeeds. After landing, place weight in aircraft to shift CG and repeat flight test again. Want to repeat series of 6 passes and landings to collect data for three different cg positions.

Preflight Prep:

— 1. Weight and Balance Full Fuel Tank
Position #1

— 2. Determine course for flight maneuvers. Prefer racetrack with long enough legs to have period of constant airspeed at trimmed condition.

— 3. Record Flight Conditions

Press:________

Temp:________

Field Elevation:________

Wind:________

— 4. Flight Recorder Checks

Maneuver Description:

Flight #1

	1. Start ground stopwatch when recorder is started.	
	2. Take Off	T/O Actual Time:_______
	3. <u>Familiarize/Warm-up</u> for pilot. After some maneuvers at his discretion, continue.	
	4. Pass #1-1 Constant airspeed, constant altitude.	Time:_______
	5. Pass #1-2 Constant airspeed, constant altitude.	Time:_______
	6. Pass #1-3 Constant airspeed, constant altitude.	Time:_______
	7. Pass #1-4 Constant airspeed, constant altitude.	Time:_______
	8. Pass #1-5 Constant airspeed, constant altitude.	Time:_______
	9. Pass #1-6 Constant airspeed, constant altitude.	Time:_______
	10. Land aircraft	Time:_______ Actual Land Time:_______
	11. Offload Data	
	12. Weight and Balance	

Remarks:

Flight #2

	1. Add weight to aircraft to shift CG	Weight Added:________
	2. Weight and Balance for Position #2	
	3. Refuel, preflight, and start ground stopwatch when recorder is started.	
	4. Take Off	Actual T/O Time:________
	5. Pass #2-4 Constant airspeed, constant altitude.	Time:________
	6. Pass #2-2 Constant airspeed, constant altitude.	Time:________
	7. Pass #2-3 Constant airspeed, constant altitude.	Time:________
	8. Pass #2-4 Constant airspeed, constant altitude.	Time:________
	9. Pass #2-5 Constant airspeed, constant altitude.	Time:________
	10. Pass #2-6 Constant airspeed, constant altitude.	Time:________
	11. Land aircraft	Time:________ Actual Land Time:________
	12. Offload Data	
	13. Weight and Balance	

Remarks:

Flight #3

	1. Weight and Balance for Position #3	
	2. Refuel, preflight, and start ground stopwatch when recorder is started.	
	3. Take Off	Actual T/O Time:_______
	4. Pass #3-1 Constant airspeed, constant altitude.	Time:_______
	5. Pass #3-2 Constant airspeed, constant altitude.	Time:_______
	6. Pass #3-3 Constant airspeed, constant altitude.	Time:_______
	4. Pass #3-1 Constant airspeed, constant altitude.	Time:_______
	8. Pass #3-5 Constant airspeed, constant altitude.	Time:_______
	9. Pass #3-6 Constant airspeed, constant altitude.	Time:_______
	10. Land aircraft	Time:_______ Actual Land Time:_______
	11. Offload Data	
	12. Weight and Balance	

Remarks:

APPENDIX D: WEIGHT AND BALANCE FORM

Flight Test Weight and Balance Worksheet FOG-R UAV Date:________

CG Position #1 - Preflight

Item	Weight (lbs) (lbs)		Arm (in)		Moment in-lb
Nose		x	8.5	=	
Left Main		x	33.375	=	
Right Main		x	33.375	=	
Total					

CG= in. %MAC

CG Position #1 - Postflight

Item	Weight (lbs) (lbs)		Arm (in)		Moment in-lb
Nose		x	8.5	=	
Left Main		x	33.375	=	
Right Main		x	33.375	=	
Total					

CG= in. %MAC

CG Position #2 - Preflight

Item	Weight (lbs) (lbs)		Arm (in)		Moment in-lb
Nose		x	8.5	=	
Left Main		x	33.375	=	
Right Main		x	33.375	=	
Total					

CG= in. %MAC

CG Position #2 - Postflight

Item	Weight (lbs) (lbs)		Arm (in)		Moment in-lb
Nose		x	8.5	=	
Left Main		x	33.375	=	
Right Main		x	33.375	=	
Total					

CG= in. %MAC

CG Position #3 - Preflight

Item	Weight (lbs) (lbs)		Arm (in)		Moment in-lb
Nose		x	8.5	=	
Left Main		x	33.375	=	
Right Main		x	33.375	=	
Total					

CG= in. %MAC

CG Position #3 - Postflight

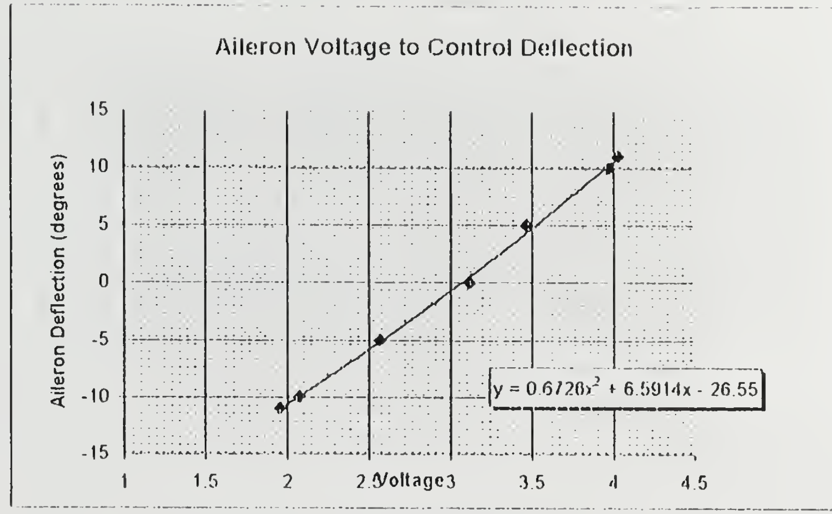
Item	Weight (lbs) (lbs)		Arm (in)		Moment in-lb
Nose		x	8.5	=	
Left Main		x	33.375	=	
Right Main		x	33.375	=	
Total					

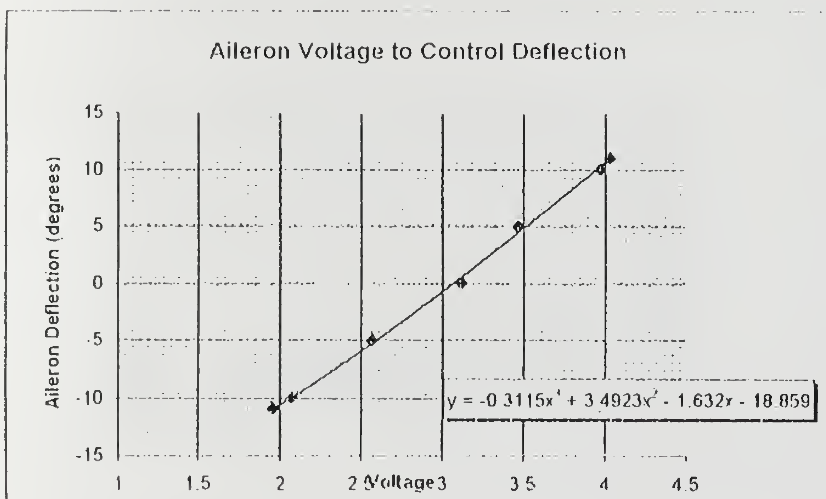
CG= in. %MAC

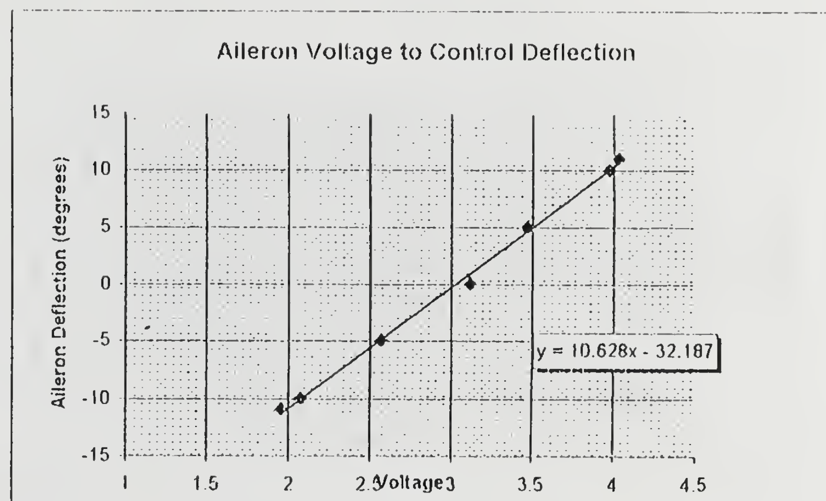
APPENDIX E: CALIBRATION DATA

Data Collected

Voltage	Deflection(deg)	
4.03	11	RTE up
3.97	10	
3.47	5	
3.12	0	
2.57	-5	
2.08	-10	
1.96	-11	RTE down

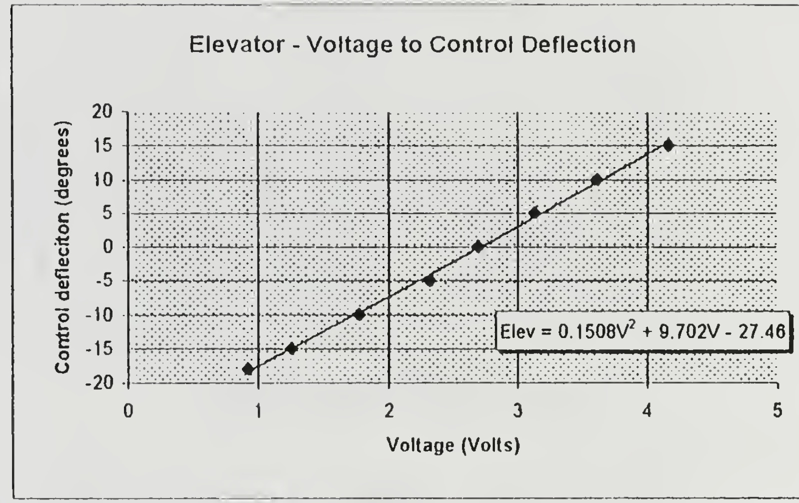


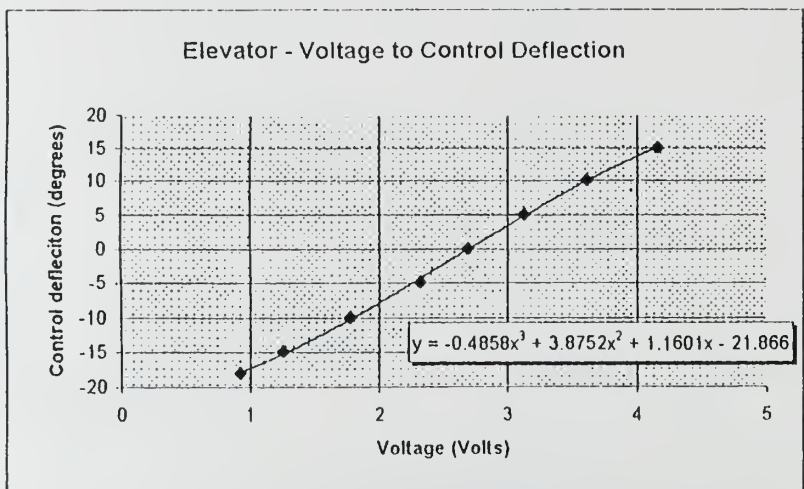


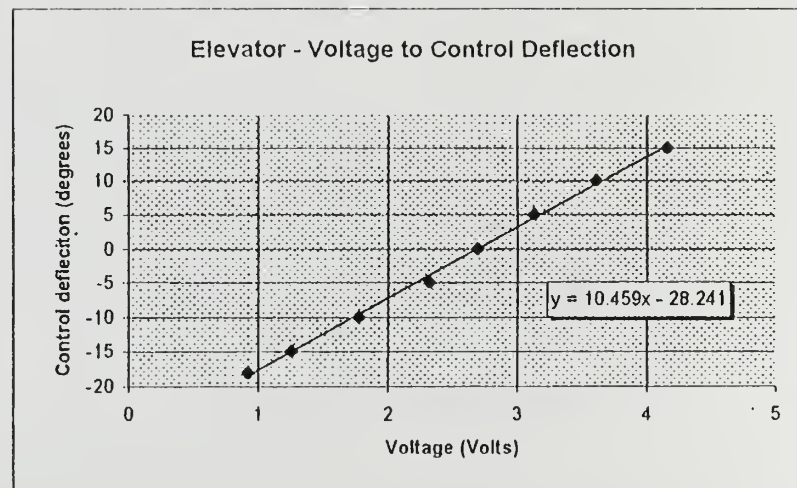


Data Collected

Voltage	Deflection(deg)	
4.16	15	TE down
3.61	10	
3.13	5	
2.69	0	
2.32	-5	
1.78	-10	
1.26	-15	
0.93	-18	TE up

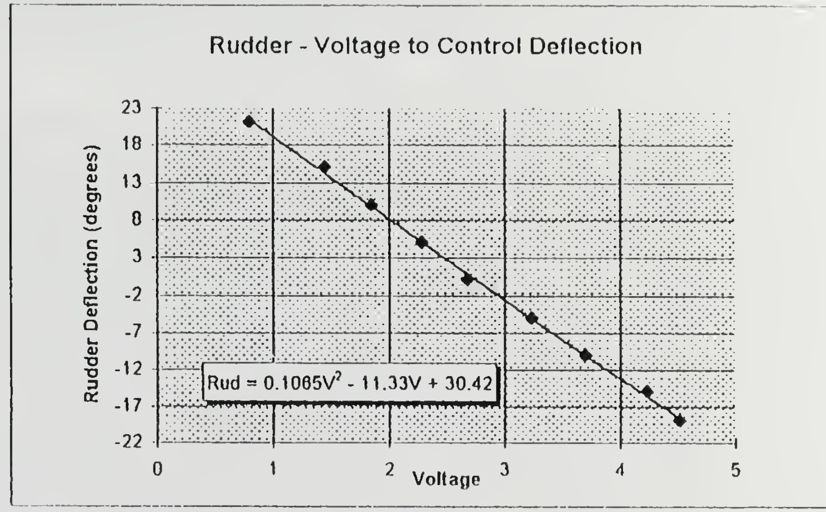


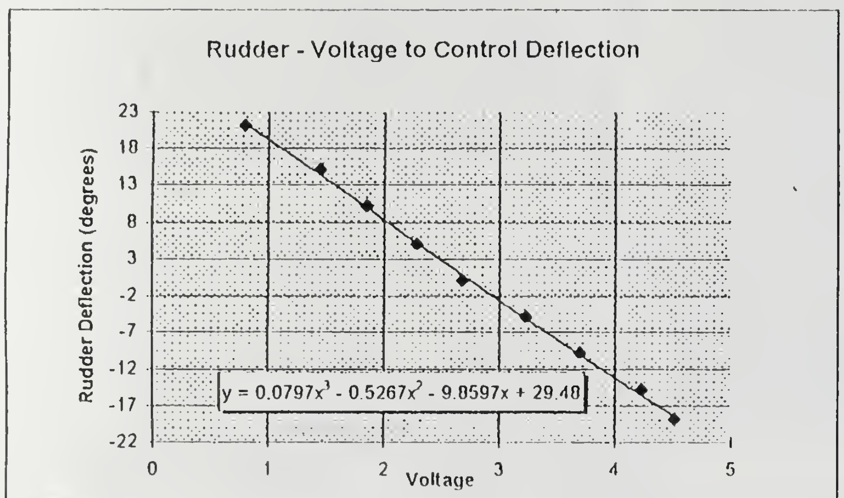


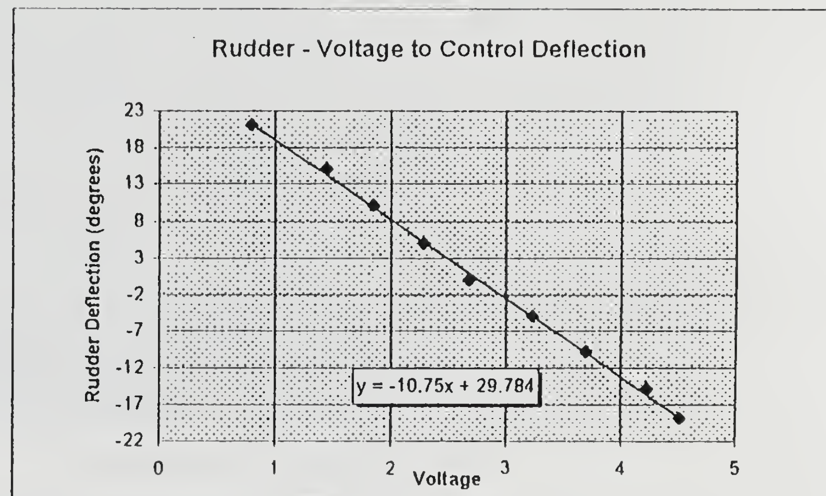


Data Collected

Voltage	Deflection(deg)	
0.8	21	TE left
1.45	15	
1.85	10	
2.29	5	
2.68	0	
3.23	-5	
3.7	-10	
4.23	-15	
4.52	-19	TE right

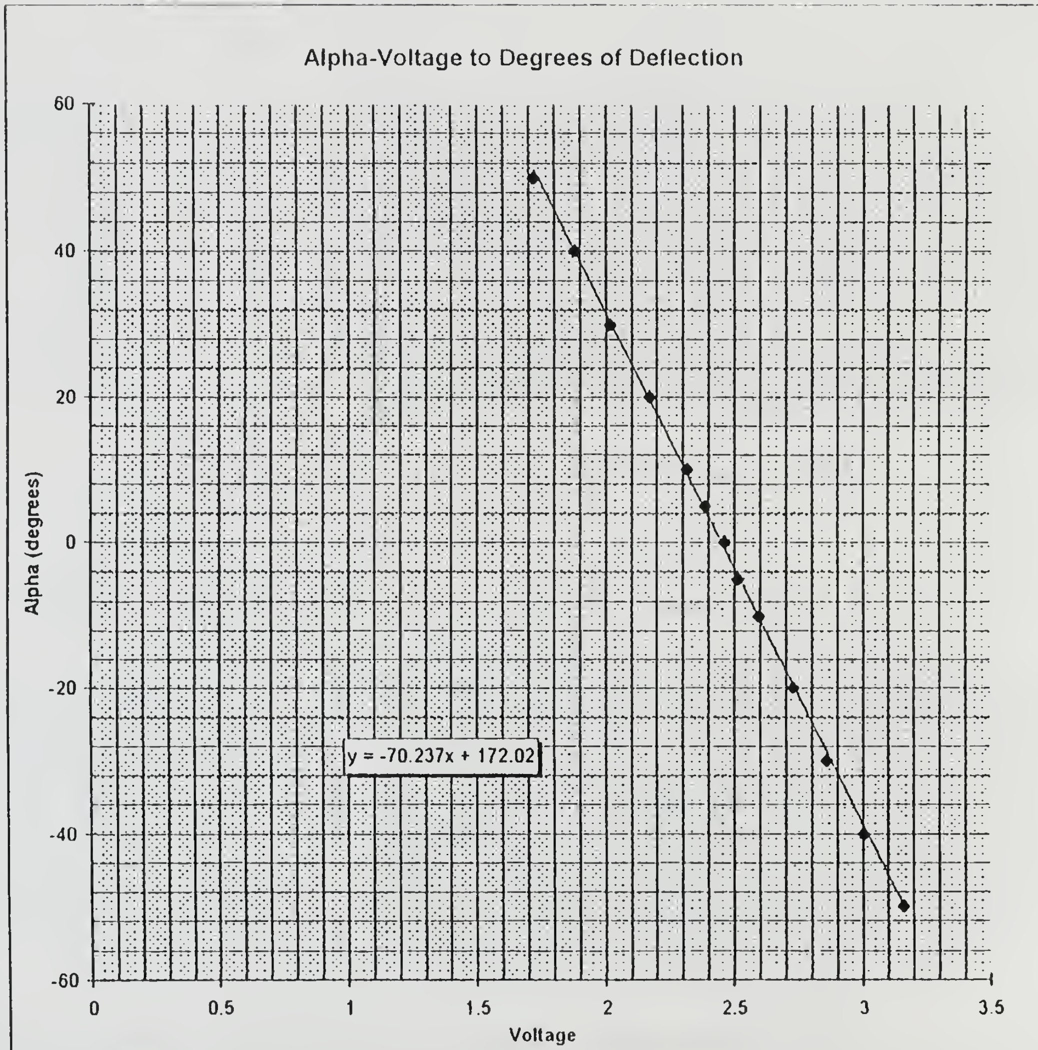






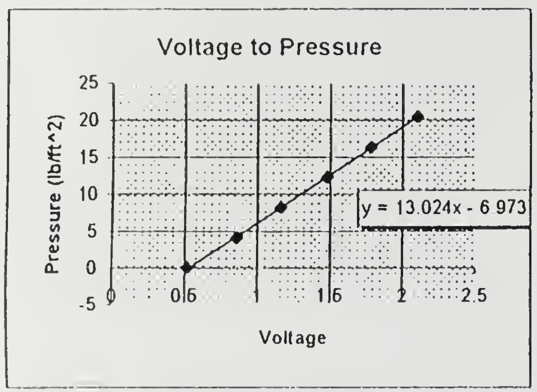
Data Collected

Voltage	Deflection(deg)	
1.721	50	Nose up
1.882	40	
2.021	30	
2.172	20	
2.318	10	
2.389	5	
2.464	0	
2.514	-5	
2.595	-10	
2.73	-20	
2.86	-30	
3.007	-40	
3.161	-50	Nose down



Data Collected

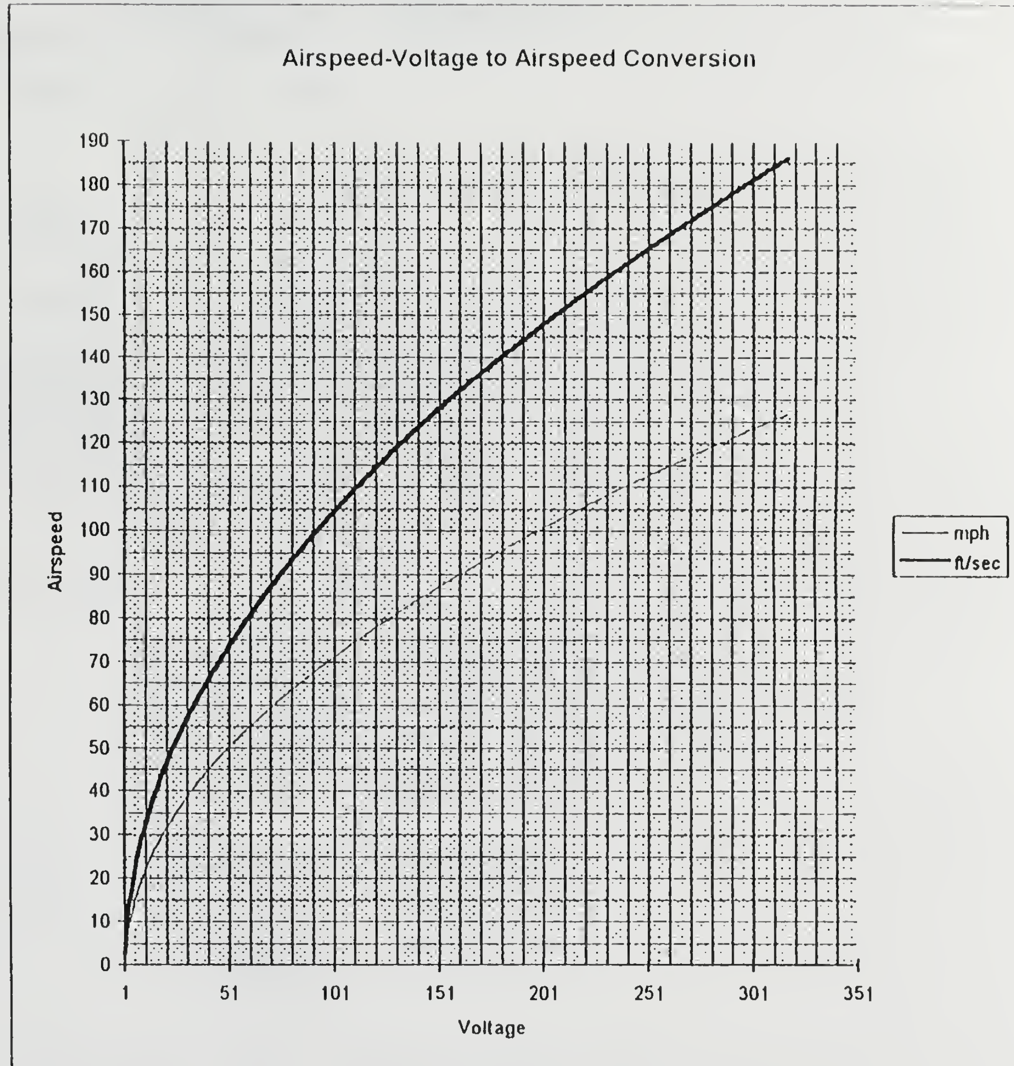
Voltage	Pressure (cm of H2O)	(lbs/ft^2)	Ve
0.554	0	0	0
1.03	5	10.22868	92.77252
1.489	10	20.45736	131.2002
2.24	15	30.68604	160.6867
0.524	0	0	0
0.863	2	4.091472	58.67449
1.166	4	8.182944	82.97826
1.48	6	12.27442	101.6272
1.786	8	16.36589	117.349
2.106	10	20.45736	131.2002



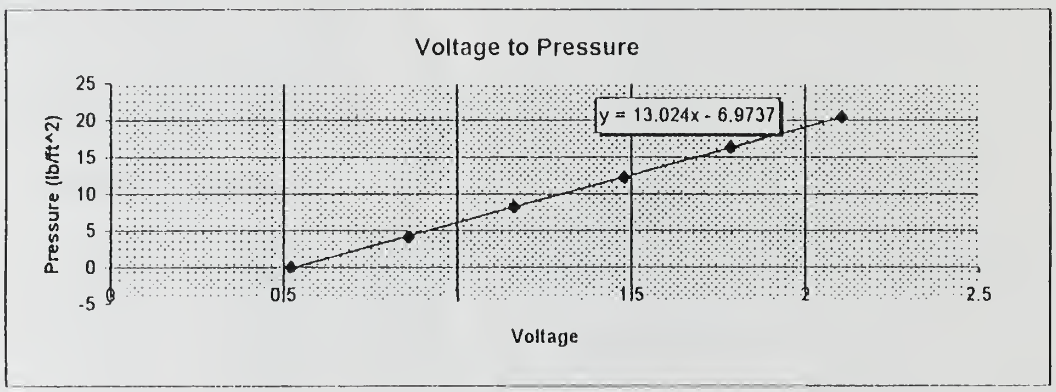
row0= 2.38E-03

$$V_e = \sqrt{\frac{2 \cdot P}{\rho_0}}$$

Voltage	Pressure (lb/ft^2)	Ve (ft/sec)	Ve (mph)
0.536	0.007164	2.455203	1.674002
0.54	0.05926	7.061393	4.814586
0.55	0.1895	12.62741	8.609595



	A	B	C	D	E
1	Data Collected				
2		Pressure			
3	Voltage	(cm of H2O)	(lbs/ft^2)	Ve	
4	0.554	0	=B4*0.0328*62.37	=SQRT(2*C4/row0)	
5	1.03	5	=B5*0.0328*62.37	=SQRT(2*C5/row0)	
6	1.489	10	=B6*0.0328*62.37	=SQRT(2*C6/row0)	
7	2.24	15	=B7*0.0328*62.37	=SQRT(2*C7/row0)	
8					
9	0.524	0	=B9*0.0328*62.37	=SQRT(2*C9/row0)	
10	0.863	2	=B10*0.0328*62.37	=SQRT(2*C10/row0)	
11	1.166	4	=B11*0.0328*62.37	=SQRT(2*C11/row0)	
12	1.48	6	=B12*0.0328*62.37	=SQRT(2*C12/row0)	
13	1.786	8	=B13*0.0328*62.37	=SQRT(2*C13/row0)	
14	2.106	10	=B14*0.0328*62.37	=SQRT(2*C14/row0)	



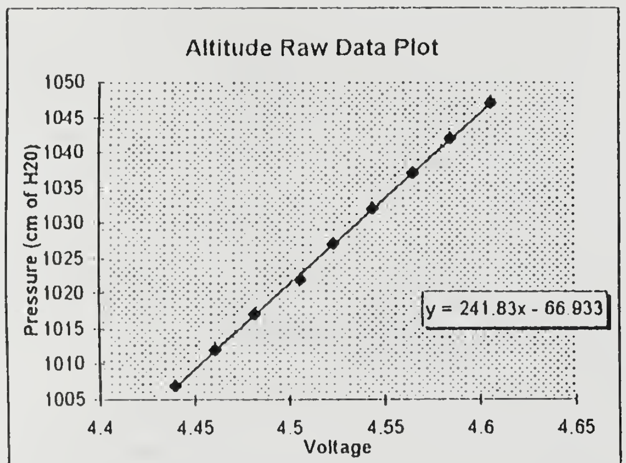
	A	B	C	D	E
15–27					
28	row0=	0.0023769			
29					
30					
31–33	$V_e=\sqrt{\frac{2\cdot P}{\rho_0}}$				
34	Voltage	Pressure (lb/ft^2)	Ve (ft/sec)	Ve (mph)	
35	0.536	=13.024*A35-6.9737	=SQRT(2*B35/row0)	=C35*0.68181818	
36	0.54	=13.024*A36-6.9737	=SQRT(2*B36/row0)	=C36*0.68181818	

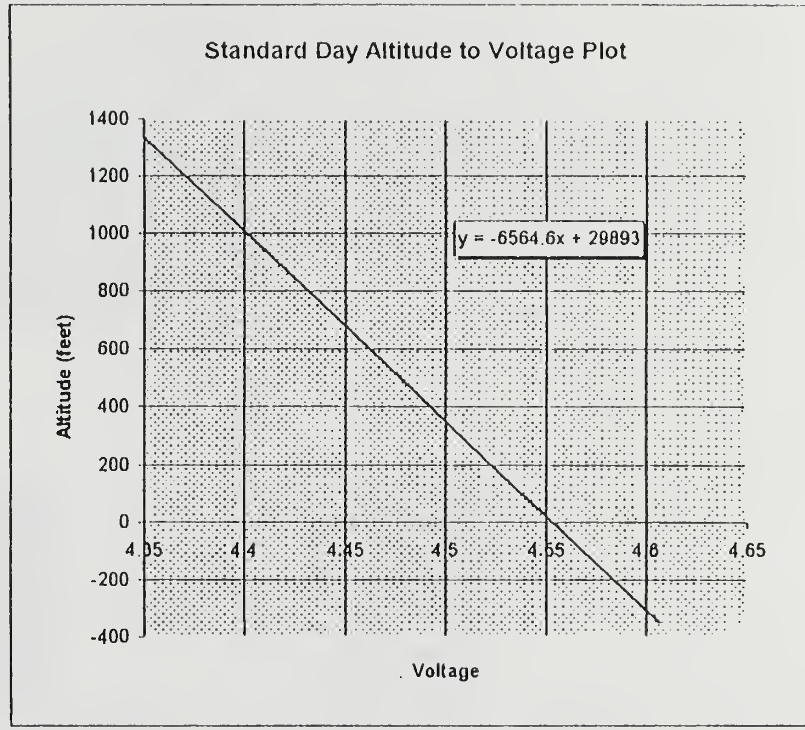
Data Collected

Barometer:

in HG	millibars	cm H20	lb/ft^2
30	1015.9164	1036.983	2121.786

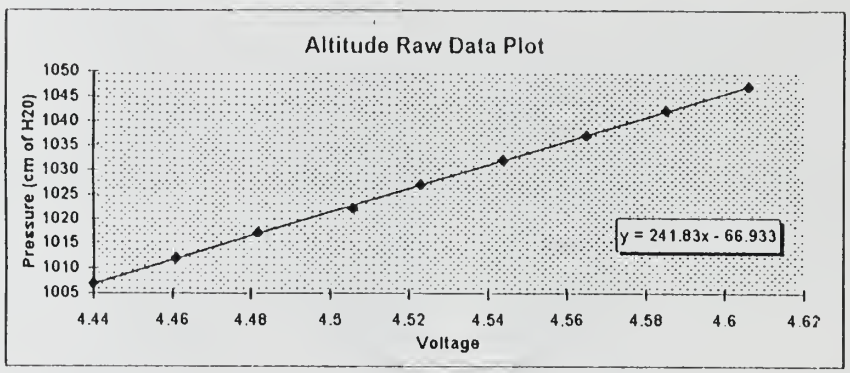
Voltage	Pressure Change (cm H20)	Absolute (cm H20)	Pressure (lbs/ft^2)
4 606	10	1046.98271	2141.850222
4 585	5	1041.98271	2131.621542
4.565	0	1036.98271	2121.392862
4.544	-5	1031.98271	2111.164182
4.523	-10	1026.98271	2100.935502
4.506	-15	1021.98271	2090.706822
4.482	-20	1016.98271	2080.478142
4.461	-25	1011.98271	2070.249462
4.44	-30	1006.98271	2060.020782





$$H = \frac{1 - \left(\frac{P_a}{P_0}\right)^{\frac{1}{5.2561}}}{6.87535 \cdot 10^{-6}}$$

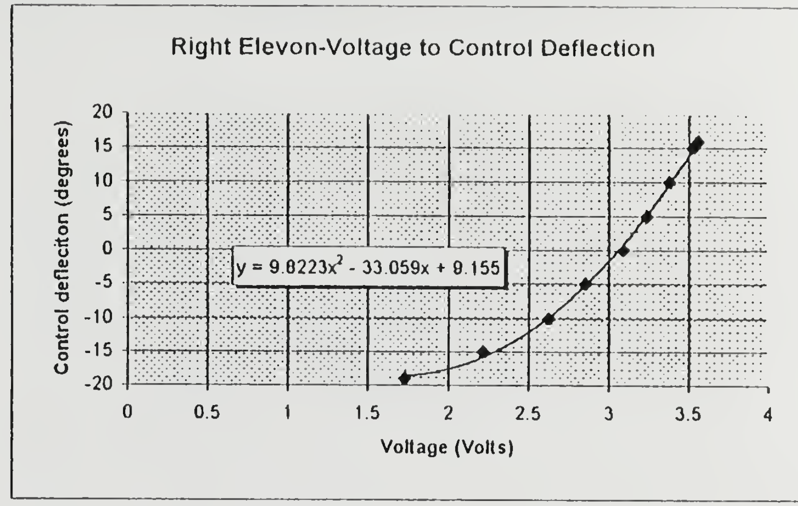
	A	B	C	D
1				
2	Data Collected			Barometer:
3			in HG	millibars
4			30	=C4*33.86388
5				
6		Pressure Change	Absolute	Pressure
7	Voltage	(cm H20)	(cm H20)	(lbs/ft^2)
8	4.606	10	=B8+$E$4	=C8*0.0328*62.37
9	4.585	5	=B9+$E$4	=C9*0.0328*62.37
10	4.565	0	=B10+$E$4	=C10*0.0328*62 37
11	4.544	-5	=B11+$E$4	=C11*0.0328*62.37
12	4.523	-10	=B12+$E$4	=C12*0.0328*62.37
13	4.506	-15	=B13+$E$4	=C13*0.0328*62.37
14	4.482	-20	=B14+$E$4	=C14*0.0328*62.37
15	4.461	-25	=B15+$E$4	=C15*0.0328*62.37
16	4.44	-30	=B16+$E$4	=C16*0.0328*62.37
17				
18				
19				
20				
21				
22				
23				
24				
25				
26				
27				
28				
29				
30				
31				
32				
33				
34				
35				
36	Voltage	Press (cmH2O)	Press (lb/ft^2)	Altitude (ft)
37	4.607	=241.83*A37-66.933	=B37*2.04611472	=(1-(C37/2116.22)^(1/5.2561))/0.00000687535
38	4.606	=241.83*A38-66.933	=B38*2.04611472	=(1-(C38/2116.22)^(1/5.2561))/0.00000687535
39	4.605	=241.83*A39-66.933	=B39*2.04611472	=(1-(C39/2116.22)^(1/5.2561))/0.00000687535



$$H = \frac{1 - \left(\frac{P_a}{P_0}\right)^{\frac{1}{5.2561}}}{6.87535 \cdot 10^{-6}}$$

Data Collected

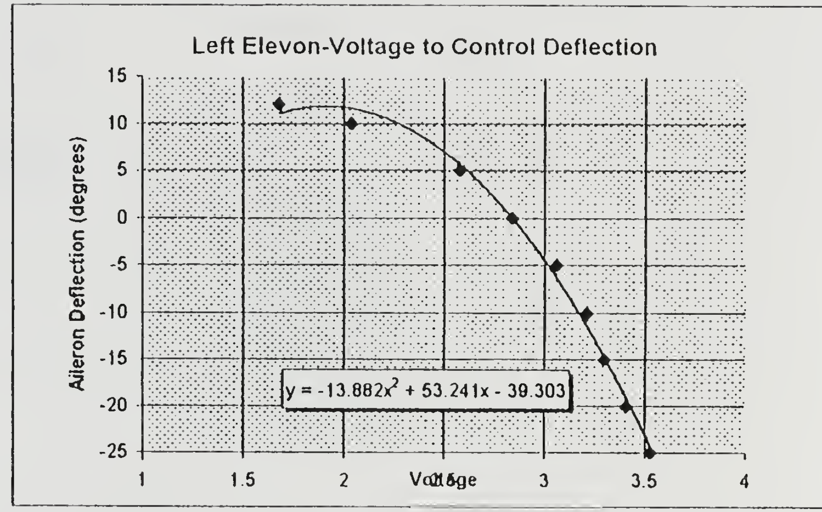
Voltage	Deflection(deg)	
3.56	16	TE down
3.53	15	
3 38	10	
3.24	5	
3.09	0	
2.86	-5	
2 63	-10	
2.22	-15	
1.73	-19	TE up



Data Collected

Voltage	Deflection(deg)	
1.68	12	TE down
2.04	10	
2.58	5	
2.84	0	
3.06	-5	
3.21	-10	
3.3	-15	
3.41	-20	
3.53	-25	TE up

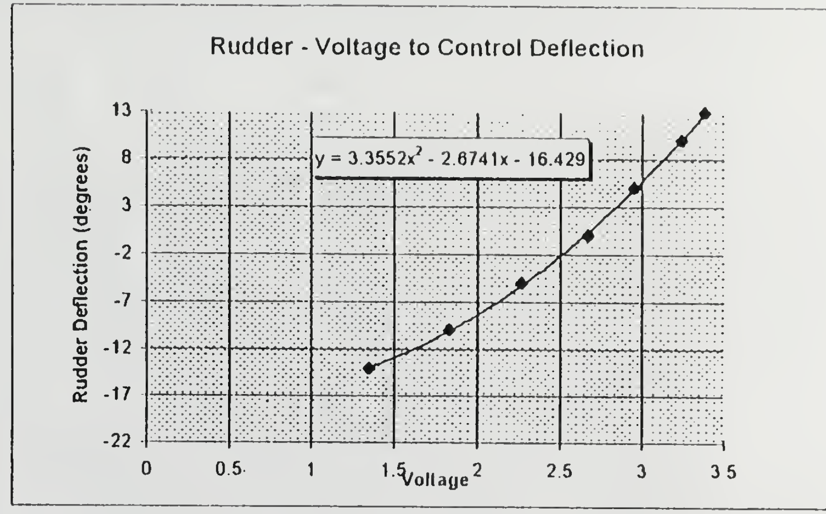
Note: Convention TE up is negative.



Data Collected

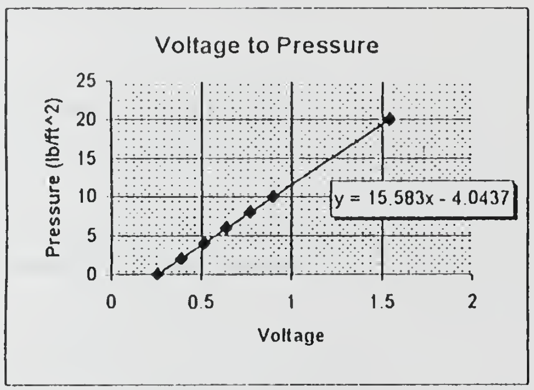
Voltage	Deflection(deg)	
3.38	13	TE left
3.24	10	
2.95	5	
2.67	0	
2.27	-5	
1.83	-10	
1.35	-14	TE right

Note: Convention TE left is positive



ata Collected

Voltage	Pressure (cm of H2O)	(lbs/ft^2)	Ve
0.26	0	0	0
0.39	2	4.091472	58.67449
0.52	4	8.182944	82.97826
0.64	6	12.27442	101.6272
0.77	8		
0.9	10	20.45736	131.2002
1.545	20	40.91472	185.545

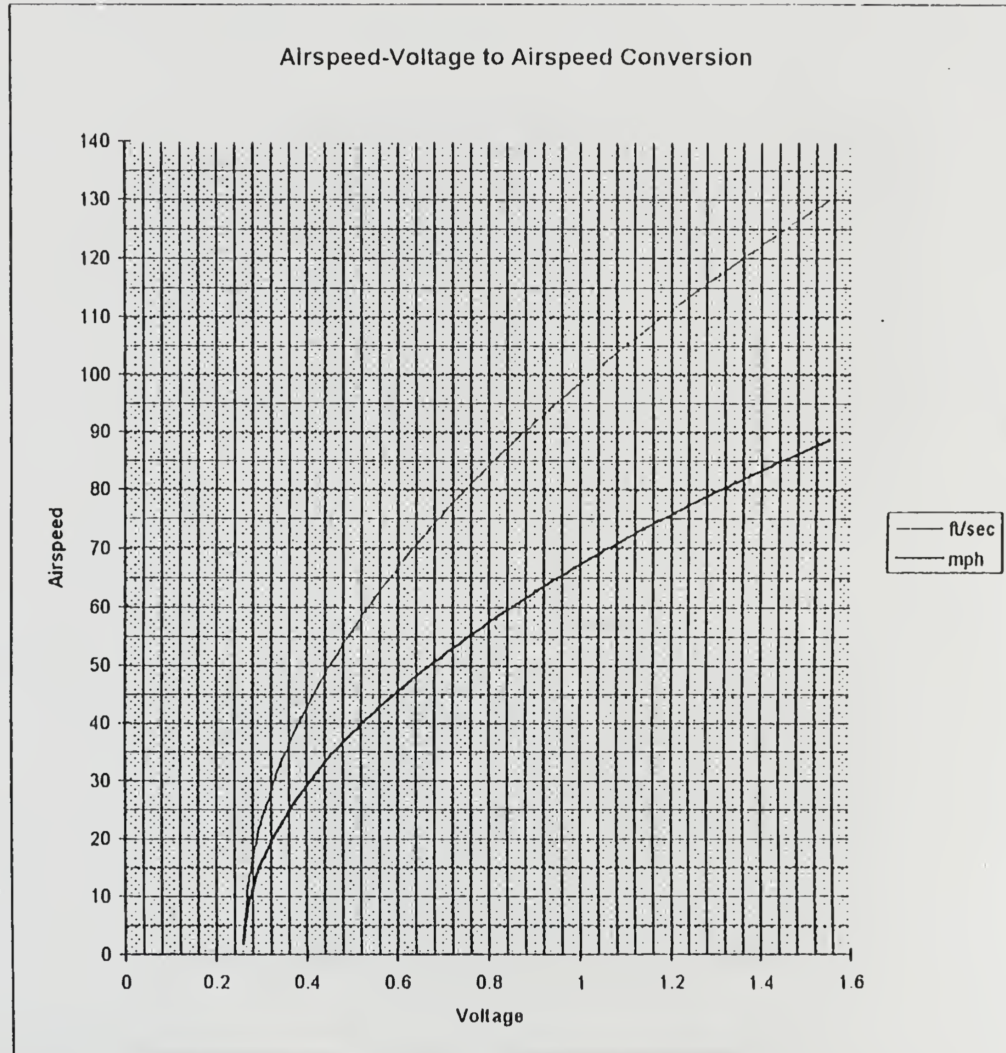


row0= 2.38E-03

NOTE: Use the numbers above for coefficients in senscals

$$V_e = \sqrt{\frac{2 \cdot P}{\rho_0}}$$

Voltage	Pressure (lb/ft^2)	Ve (ft/sec)	Ve (mph)
0.26	0.00788	2.574973	1.755663
0.27	0.16371	11.73673	8.002316

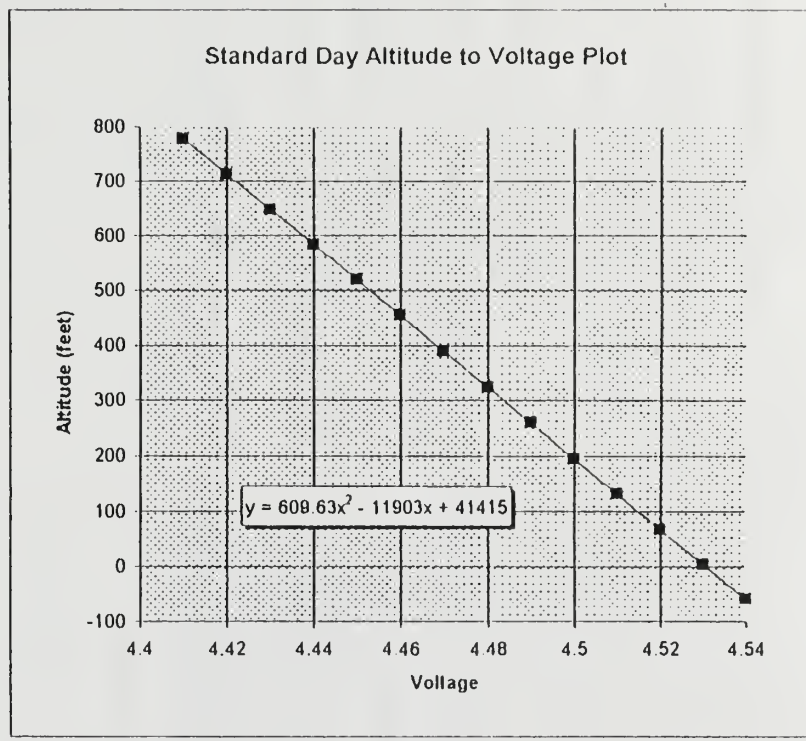


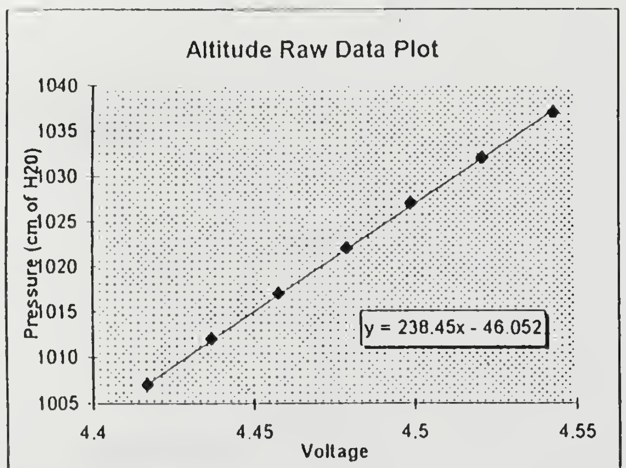
Data Collected

Barometer:

in HG	millibars	cm H20	lb/ft^2
30	1015.9164	1036.983	2121.786

Voltage	Pressure Change (cm H20)	Absolute (cm H20)	Pressure (lbs/ft^2)
4.543	0	1036.98271	2121.392862
4.521	-5	1031.98271	2111.164182
4.499	-10	1026.98271	2100.935502
4.479	-15	1021.98271	2090.706822
4.458	-20	1016.98271	2080.478142
4.437	-25	1011.98271	2070.249462
4.417	-30	1006.98271	2060.020782



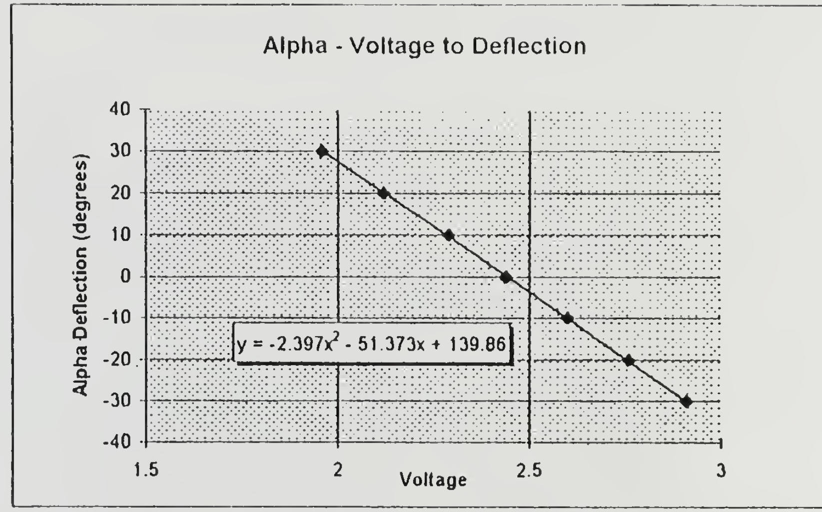


$$H = \frac{1 - \left(\frac{P_a}{P_0}\right)^{\frac{1}{5.2561}}}{6.87535\,10^{-6}}$$

Data Collected

Voltage	Deflection(deg)	
1.96	30	Nose Up
2.12	20	
2.29	10	
2.44	0	
2.6	-10	
2.76	-20	
2.91	-30	Nose Down

Note: Nose up is positive



APPENDIX F: DATA RECORDER OPERATIONS

A. DESCRIPTION

The flight data recording system is set up to record flight data for Unmanned Air Vehicles. It is capable of recording up to eight channels of data at six samples per second for 50 minutes total time. The data is then downloaded from the data recorder via a RS-232 serial connection for analysis.

The UAV is wired with a wiring harness coming from a maximum of eight data sensors onboard the aircraft to a 25-pin plug which connects to the data recorder. The recorder is installed prior to going to the field. On the control panel of the aircraft is a switch, for turning the recorder on and off, and a 1/4-inch plug for connecting to the computer.

B. CONCEPT OF OPERATIONS

The data recorder is made to record flight test data from a UAV. The recorder must be used in conjunction with a computer to download the recorded information.

Typically, in the field, an operator will have a laptop computer and the UAV equipped with the recorder. The computer is connected to the UAV to check proper recorder operation. The computer is disconnected and the UAV flown. After landing, the computer is connected to the UAV, the collected data is downloaded to a file on the computer. A tattletale program converts the data to an ASCII format. The newly formatted data can be used by software of choice for analysis.

When ready for the flight testing to begin, the computer is connected to the aircraft. The recorder is turned on. The pilot of the aircraft toggles the recording switch on his remote-control to RECORD position. The data recorder will deliver a message to the computer that it is working properly. The pilot should toggle his remote-control record switch to OFF. The computer is disconnected from the aircraft. The recorder is now ready to begin.

When ready to begin recording, the pilot toggles the switch on his remote-control to RECORD. The recorder is recording data. When finished collecting data, the pilot toggles the remote-control record switch to OFF. The computer is connected to the aircraft and the data is downloaded to the computer and stored in a file for later analysis.

While recording data, if the remote-control recorder switch is toggled to OFF, recording will stop. If it is toggled to RECORD position, the recorder will begin

recording. However, all previous data recorded will be lost. After recording has stopped, the data must be downloaded prior to recording more data.

C. FILE CONVENTIONS

Use of the UAV Data Recorder involves five different kinds of files. To keep files organized a convention has been adopted and is used by the MATLAB Tattle5F toolbox when it generates the various files. Below is a list of the files and an explanation of each.

Type of file	File Extension	Format of Contents	Description/Contents of File
Raw data file	.dat	Tattletale binary. Most significant byte first, least significant byte second.	Raw data downloaded from the data recorder.
Converted data file	.bin	Intel binary. Least significant byte first, most significant byte second	Data file created by MATLAB command, dat2bin.m from -.dat data file.
Raw ASCII data file	.txt	ASCII	Raw data saved in ASCII format. Created by MATLAB command, bin2asc.m from -.bin data file.
Reduced ASCII data file	.red	ASCII	Reduced data saved in ASCII format. Created by MATLAB command redasc.m using sensor calibrations and -txt data file.
TxBasic source file for data recorder	.txb	ASCII	TxBasic source code for programming the UAV Data Recorder.

Files offloaded from the data recorder may be given any name desired at the time of offload. However, the following was used during development and is recommended.

tnov12_1.dat

t	Test - indicates it is a test data file.
nov	Month data was downloaded.
12	Day of the month data was downloaded.
_1	Order in the series of data files downloaded for the day. i.e. if more than one file was downloaded on 12 NOV, they would have numbers 1, 2, 3, ...
.dat	File extension indicates format of data in file. this format indicates it is a Tattletale binary raw data file.

TxBasic source code files used by the data recorder use the following convention for file names.

r11nov96.txb

r	Recorder - indicates it is a data recorder source file.
11nov96	Date source code was last updated.
.txb	File extension indicates TxBasic file.

D. OFFLOADING THE DATA

CAUTION: Do NOT secure power to the data recorder before downloading data. All data will be lost if the recorder is turned off.

I. Computer - START TxTools
II. Patch Chord - CONNECT
III. Computer - [CTRL]+[C]
IV. Computer - DATA ADDRESS
   A. Listed on the screen is the End of Data (EOD) address for the data stored in the recorder. This number is required for the data offload. Write it down for later use.

V. Computer - OFFLOAD DATA
   A. In TxTools, follow menu selections below:
      1. >Tattletale > Offload data file...
      2. Start Address = 0
      3. End Address = [EOD address]
      4. >Off-load
      5. Type file name with ".dat" extension. >OK
      6. Progress bar will show status.
   B. Unsuccessful Offload - Error message will pop up. Begin offload procedures again with step (1) above.

   C. Successful Offload - Progress bar will disappear and control will return to the TxTools Terminal Window.

VI. Patch Chord - DISCONNECT

E. LOOKING AT THE DATA

MATLAB is used to reduce and view the recorder data. A MATLAB toolbox, Tattle5F, contains the files and functions required. Because the functions are set up in a toolbox, the MATLAB 'help' command may be used at any time get more details about the toolbox itself or any of its functions. Simply type 'help' followed by the command assistance is required for. For information about the toolbox itself, type 'help Tattle5F'.

The following pages contain checklists for reducing and plotting the data recorder data. Any details about particular commands may be obtained by using 'help.'

Initial Reduction and Plot of Data
This procedure should be used for the initial reduction and plotting of data recorder data. If a raw data file has already been reduced. The procedure on the following page, 'Plotting Data,' may be used. This procedure may be used to reduce data using new calibration coefficients also.

Start MATLAB
Ensure the data recorder raw data file is in the current directory.
Load the sensor calibration coefficients

>senscals

NOTE: This assumes the sensor calibration coefficients have been loaded into the senscals file of the Tattle5F toolbox. Each time a new calibration is performed, use the Windows Notepad to modify the senscals.m file in the Tattle5F toolbox. Modify the 8x3 matrix containing the sensor calibration coefficients.

Run the Reduce and Plot command

>redplot('filename.dat',a,rate)

filename.dat - the data recorder raw data file.

a - the variable name of the 8x3 matrix containing the sensor calibration coefficients. If senscals.m was run, the variable name was loaded into memory. Use that variable name as appropriate.

rate - (optional) - rate data recorder collected data. Default is 40Hz. If something other than 40Hz was used, it should be specified here.

Computer will prompt with, 'type in the number of the channels to plot e.g. [1 2 4] '. Type in an array containing the channels to plot.

>[0 1 3]

NOTE: type in any array of numbers from 0 to 7. The maximum channels which can be plotted on one chart is 6. A single channel may be plotted also.

Chart will be plotted.

To reduce range of plot.

>Use mouse to set crosshair at corner of imaginary box surrounding desired range.
>Click left mouse key.

>Set crosshair at opposite corner of imaginary box surrounding desired range.
>Click left mouse key.

Chart will be replotted

To reduce range of plot, repeat above procedures.

When happy with plot.

>Set crosshair on chart.
>Click right mouse button.

Plotting Data

This procedure can be used to view data saved in an ASCII format. Both -.txt and -.red files may be plotted. If the original data recorder -.dat, raw data file has not been reduced using redplot.m command, the procedure, 'Initial Reduction and Plot of Data,' should be used.

Run MATLAB
Run the Plot ASCII function.
>plotasc('filename.red')

filename.red - (optional) - This is the reduced ASCII data file. A file with -.txt extension may also be used. If no filename is specified, a prompt will allow browsing for the desired file to plot.

Computer will prompt with, 'type in the number of the channels to plot e.g. [1 2 4] '. Type in an array containing the channels to plot.

>[0 1 3]

NOTE: type in any array of numbers from 0 to 7. The maximum channels which can be plotted on one chart is 6. A single channel may be plotted also.

Chart will be plotted.

To reduce range of plot.

>Use mouse to set crosshair at corner of imaginary box surrounding desired range.
>Click left mouse key.

>Set crosshair at opposite corner of imaginary box surrounding desired range.
>Click left mouse key.

Chart will be replotted

To reduce range of plot, repeat above procedures.

When happy with plot.

>Set crosshair on chart.
>Click right mouse button.

F. CHANGING SAMPLING RATE

Changing the sampling rate of the data recorder involves changing some TxBasic code lines. The sampling rate is set with a combination of a RATE command and a SLEEP command. The syntax for these lines is shown below.

RATE R

SLEEP S

R - an integer up to 255 which is a factor of 192.

S - an integer.

The table below shows combinations R and S which will yield the given sampling rates.

	S				
R	**1**	**2**	**3**	**4**	**5**
1	100	50	33.3	25	20
2	200	100	66.67	50	40
3	300	150	100	75	60
4	400	200	133.3	100	80
6	600	300	200	150	120
8	800	400	266.67	200	160
12	1200	600	400	300	240
16	1600	800	533.33	400	320
24	2400	1200	800	600	480
32	3200	1600	1066.67	800	640

Other sampling rates may be obtained by following this formula.: $\mathrm{SampleRate} = \frac{R \cdot 100}{S}$

Note: R must be a factor of 192 or the sample rate will be inaccurate.

Following is a list of files and the command lines which should be changed for the data recorder and the MATLAB toolbox, Tattle5F to work properly.

TxBasic File

- r24nov96.txb (most current TxBasic Program)

⇒ print "Recording at 40 samples/sec." print	Change the '40' to the appropriate rate.
⇒ rate 2	Change the '2' to the appropriate number based on the table above.
⇒ sleep 0 while pin(0) > 0	DO NOT CHANGE
⇒ sleep 5 burst dfPoint,8,2 cnt = cnt + 1	Change the '5' to the appropriate number based on the table above.
⇒ if cnt%40=0 print cnt/40," "	Change the two 40's to the appropriate rate. This controls the printing on the terminal screen.

MATLAB Toolbox Files

The MATLAB files do no need to be modified. However, for rates other than 40 Hz, the rate must be given as an argument for the MATLAB functions. See the individual MATLAB functions for details.

G. CHANGING CALIBRATION COEFFICIENTS

In the MATLAB Tattle5F toolbox, there is a file called senscals.m. This file simply defines the matrix variables which have the calibration coefficients for a particular aircraft. In the current senscals.m file, the two variables shown below are defined.

Variable	Description
lf	LoFlyte coefficients
F	FOG-R coefficients

This file is NOT required to run any of the toolbox functions. However, it is required to supply some of the toolbox functions with an 8x3 matrix variable. This can be done with senscals.m, with some other file, or by simply typing it directly in at the MATLAB command window. To maintain/change senscals.m, use Windows Notebook or some other text editor and change the matrices in the file.

NOTE:

Keep in mind the following points when defining the matrix variable by any means

- The matrix should be 8x3. This corresponds to eight channels of the recorder in ascending order from channel (0) to channel (7). Each channel has three coefficients which come from a second order equation describing the calibration relation for a particular channel.
- The number of rows of the matrix should match the number of channels recorded. See "Changing Number of Channels Recorded" for more about how to change the number of channels recorded.

- Normally, the coefficients from a second-order equation describing the voltage to parameter conversion are used in data reduction. However, because the voltage to airspeed conversion is not accurately described by a second-order equation when using a pressure sensor, the Tattle5F toolbox commands handle the airspeed data reduction differently than other data reductions. The coefficients from a linear equation descibing the conversion from voltage to pressure are used. In the 8x3 matrix variable used to pass the calibration coefficients to the MATLAB function, the first element of the row with airspeed coefficients is 999. The remaining elements of the row have the two coefficents from the linear equation describing the voltage to pressure conversion. The 999 tells the MATLAB function to handle that particular row of coefficients specially by doing unique calculations for airspeed.
- For details of how the data is reduced, examine the MATLAB Tattle5F function, redasc.

H. CHANGING NUMBER OF CHANNELS RECORDED

To change the number of channels the data recorder records at one time, the following files must be modified. Make backups of the current files and then modify as shown below. The files not listed do not need to be modified.

TxBasic Program

- r24nov96.txb (most current TxBasic Program)

⇒ burst dfPoint,8,2 — Change the '8' to the desired number of channels command in Tattletale manual section five for more details.

MATLAB Tattle5F Toolbox Files

- redasc

⇒ numch=8; — Change to the number of channels.

⇒ dat=fscanf(fid,'%6g %6g %6g %6g %6g %6g %6g %6g\n',[8,8000]);

The number of "%6g" s should match the number of channels. (note: the last one should be "%6g\n")

The "8" in "[8,8000]" should be changed to the number of channels.

⇒ fprintf(fid2,'%6.3f %6.3f %6.3f %6.3f %6.3f %6.3f %6.3f %6.3f\n',dat);

The number of "%6.3f" s should match the number of channels. (note: the last one should be "%6.3f\n")

I. SETTING UP NEW COMPUTER

This section will give a general explanation about how to set up a computer for use with the UAV data recorder, TxTools, and the MATLAB Tattle5F toolbox.

Things needed:

⇒ TxTools Disk - The most current versions of TxTools may be obtained from the Onset Computer Corporation Internet home page.

⇒ MATLAB Tattle5F Toolbox Disk - This should contain six files. One for each of the five functions and one with the help file.

TxTools Install

1. On the computer you are setting up, create a directory called TxTools.
2. Copy the files from the TxTools Disk into the directory you created.
3. To start TxTools, execute the txtools.pif file. This will start TxTools in a window.
4. Using Windows, a program group and program icon can be set up.

MATLAB TOOLBOX INSTALL

The MATLAB toolbox must be manually installed.

1. Create a new directory titled, "Tattle5F," in the MATLAB\toolbox directory.
2. Copy the six files of the Tattle5F toolbox into the newly created directory.
3. Use Windows notebook program to modify the matlabrc.m file in the MATLAB directory.

   ⇒ Look at the format of the path for the other toolboxes. Follow the format and type in the path of the new "Tattle5F" toolbox.

⇒ Save the file.

4. The toolbox is installed.

J. CHANGING EEPROM

The procedures for changing the EEPROM are in the Tattletale Operations Manual, Section 4. The instructions are straight forward. However, below are a couple of points and recommendations to keep in mind.

⇒Power to the Tattletale must be 12±0.6 VDC. It will not work with any other power supply.

⇒The new program being loaded into the EEPROM should have a file name which follows the convention explained in the "File Conventions" section above.

⇒The header of the program should contain the date the file was updated/loaded. This shows up when the recorder is turned on, providing a means for the user to know the version of the program being run.

⇒Prior to saving the "remind" file (see Tattletale Manual Section 4), load the new program into the Tattletale RAM. After it is in the RAM, continue with the procedures in the manual for saving the "remind" file.

APPENDIX G: TxBasic Program

```
        model 510
        dfPoint = 0
        onerr fulmem


        print
        print "UAV Data Recorder"
        print "Version 24 NOV 1996"
        print
        print
        print "**********************************"
        print
        print "Tattletale is receiving power."
        print "Program is running properly."
        print
        print "Toggle remote control switch to RECORD."
        print


init:   if pin(0) > 0 goto onmsg
        goto init

onmsg:  print "**********************************"
        print
        print "Recorder radio switch toggled on properly."
        print
        print "Toggle remote control switch to OFF."
        print
        print "**********************************"
        print

bgnprg:     if pin(0) > 0 goto bgnprg

        print "Recorder radio switch toggled off properly."
        print
        print "CHECKS COMPLETE"
        print
        print "Recording will begin when the transmitter switch"
        print "is toggled to RECORD...."
        print
```

```
wait:   if pin(0) > 0 goto rcrd
        cbreak finish
        goto wait

rcrd:   dfPoint = 0
        cnt = 0
        print "Recording at 40 samples/sec."
        print
        rate 2
        sleep 0
        while pin(0) > 0
          sleep 5
          burst dfPoint,8,2
          cnt = cnt + 1
          if cnt%40=0 print cnt/40," "
        wend

        goto wait

fulmem: if pin<0> > 0 goto fulmem
        goto wait

finish: print
        print "End of Data Pointer = ", dfPoint-1
```

LIST OF REFERENCES

1. Davidson, Peggy T., and Foch, Richard J., "UAV Development & Test Capabilities at the Naval Research Laboratory,", *AUVS '95 Proceedings Manual*, 1995.

2. Kenneth, Munson, *Janes Unmanned Aerial Vehicles and Targets*, Coulfdon, UK; Janes Information Group, 1995.

3. de Arcangelis, Mario, *Electronic Warfare*, Blandford Press, 1985.

4. Commander, Naval Air Systems Command, *Test and Evaluation Master Plan No. 1217, Short Range Unmanned Air Vehicle (Pioneer),* 30 October 1987.

5. Thompson, Mark, "Contract Bypassed U.S. Rules," *San Jose Mercury News*, 2 July 1988.

6. Lyons, Daniel F., *Aerodynamic Analysis of a U.S. Navy and Marine Corps Unmanned Air Vehicle*, Master's Thesis, Naval Postgraduate School, Monterey, California, June 1989.

7. Watkiss, Eric, J., *Flight Dynamics of an Unmanned Aerial Vehicle*, Master's Thesis, Naval Postgraduate School, Monterey, California, March 1994.

8. "Gulf War Experience Sparks Review of RPV Priorities," *Aviation Week and Space Technology*, April 22, 1991.

9. Metzger, Robert J., "Scannerless Imaging Laser Radar for Unmanned Air Vehicle Applications,", *AUVS '95 Proceedings Manual*, 1995.

10. Salmons, James D., *Developmental Flight Testing of a Half-Scale Unmanned Air Vehicle*, Master's Thesis, Naval Postgraduate School, Monterey, California, September, 1990.

11. Aitcheson, Kent R., *Stability and Control Flight Testing of a Half-Scale Pioneer Remotely Piloted Vehicle*, Master's Thesis, Naval Postgraduate School, Monterey, California, September 1991.

12. *Tattletale Model 5F Installation and Operation Manual*, Onset Computer Corporation, 1995.

13. *"PEGASUS" Briefing for OSD UAV Officials and OPTEC PEGASUS Working Group on Development Milestone VI*, Test and Experimentation Command Experimentation Center (TEC), September 17, 1992.

14. *Operator's Handbook for the FOG-R and Unmanned Aerial Vehicle (UAV) Instrumentation System*, 1992.

15. Solid State Pressure Sensors Handbook, Volume 15, Sensym Inc., Milipitas, California, 1996.

16. Fendley, Michael W, *Construction and Low Speed Longitudinal Testing of a Waverider Configuration*, Master's Thesis, Naval Postgraduate School, Monterey, California, December 1996.

INITIAL DISTRIBUTION LIST

No. of Copies

1. Defense Technical Information Center 2
   8725 John J. Kingman Rd., STE 0944
   Ft Belvoir, VA 22060-6218

2. Dudley Knox Library 2
   Naval Postgraduate School
   411 Dyer Rd.
   Monterey, CA 93943-5101

3. Chairman, AA 1
   Department of Aeronautics and Astronautics
   Naval Postgraduate School
   Monterey, CA 93943-5106

4. Professor Richard M. Howard, AA/Ho 3
   Department of Aeronautics and Astronautics
   Naval Postgraduate School
   Monterey, CA 93943-5106

5. Professor Conrad F. Newberry, AA/Ne 2
   Department of Aeronautics and Astronautics
   Naval Postgraduate School
   Monterey, CA 93943-5106

6. LT Joseph M. Merola 2
   25 Clark St.
   Painted Post, NY 14870

7. Robert M. Pap 1
   Accurate Automation Corporation
   7001 Shallowford Road
   Chattanooga, TN 37421

8. Joachim Beer 1
   3031 Sunrise Ave.
   Marina, CA 93933